	animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

This example demonstrates how to animate the size of an element using "Longhorn" markup language (code-named "XAML"). There are multiple ways to animate the size of an element: directly animate the height and width attributes of the element, or apply an animated ScaleTransform to the element. In this example, two Rectangle elements are resized using these methods. One rectangle is resized by animating its RectangleWidth attribute and another is resized by animating a ScaleTransform applied to the rectangle. Each rectangle is filled with a pattern to highlight the differences between the two resizing methods. Initially, the two patterns look the same, but as the rectangles are resized, patterns change depending on how their containing rectangle is resized.

In the first example, a Rectangle element's RectangleWidth property is animated using a LengthAnimationCollection and a LengthAnimation. The LengthAnimation object in this example animates the rectangle's RectangleWidth from its base value of 200 To a destination value of 600 over a Duration of 10 seconds.

```
<Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Rectangle
  RectangleTop="20"
  RectangleLeft="20"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Red"
  StrokeThickness="5">

  <Rectangle.Fill>
   <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
    ImageSource="help.gif" TileMode="Tile"/>
  </Rectangle.Fill>

  <Rectangle.RectangleWidth>
  <!-- Animate the Rectangle's width: -->
   <LengthAnimationCollection>
    <LengthAnimation
     To="600" Duration="10" AutoReverse="true" RepeatCount="50" />
   </LengthAnimationCollection>
  </Rectangle.RectangleWidth>

 </Rectangle>
```

When the previous "XAML" is run, more of the pattern is exposed as the rectangle expands; however, the question marks that make up the pattern do not grow larger.
In the next example, a TransformDecorator is used to apply a ScaleTransform to a rectangle. A DoubleAnimation is used to animate the ScaleTransform object's ScaleX value using the ScaleXAnimations attribute. The DoubleAnimation animates the ScaleX value From 1 To a destination value of 3 over a Duration of 10 seconds. As a result, the rectangle's width is scaled from 100 percent (its starting size) to 300 percent over ten seconds.

```
  <TransformDecorator AffectsLayout="False">
   <TransformDecorator.Transform>

   <!-- Use the ScaleTransform to enlarge the rectangle -->
    <ScaleTransform ScaleX="1" ScaleY="1">
     <ScaleTransform.ScaleXAnimations>
      <DoubleAnimation  From="1" To="3" RepeatCount="30"
       AutoReverse="True" Begin="0" Duration="10" />
     </ScaleTransform.ScaleXAnimations>
    </ScaleTransform>

   </TransformDecorator.Transform>

   <Rectangle
    RectangleLeft="20"
    RectangleTop="200"
    RectangleWidth="200"
    RectangleHeight="150"
    Stroke="Black"
    StrokeThickness="3">
     <Rectangle.Fill>
      <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
       ImageSource="help.gif" TileMode="Tile"/>
      </Rectangle.Fill>
  </Rectangle>
 </TransformDecorator>

</Canvas>
```

In the previous example, the DoubleAnimationCollection tag, <DoubleAnimationCollection>, is omitted when animating the transformation's scale factor. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ScaleXAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

When the second rectangle expands, the objects in the pattern also grow larger, unlike in the first rectangle. The pattern behaves this way because when you transform an element the entire element and its child elements are transformed. When you directly alter the size of an element, as in the case of the first rectangle, the element's children are not resized, unless their size and position are dependent on the size of their parent element.

Fl atAnimati n Class

Definition: Used to animate properties that accept a Single value.

Method	Descripti n
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from FloatTimedModifier.

CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this FloatTimedModifier Inherited from FloatTimedModifier.
Copy	Creates a copy of this FloatModifier Inherited from FloatModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from FloatTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from FloatTimedModifier.
EndIn	Schedules an interactive end time. Inherited from FloatTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from FloatTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FloatAnimation	Creates a new FloatAnimation with all properties set to their default values.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates the value of the animation at the current time.
GetValueImpl	Inherited from FloatModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from

	Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a FloatModifier Inherited from FloatModifier.
Pause	Pauses this timeline. Inherited from FloatTimedModifier.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from FloatTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from FloatTimedModifier.
SetDefaultParentTimeline	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from FloatTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from FloatTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from FloatTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from FloatTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from FloatTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from FloatTimedModifier.

Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from FloatTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from FloatTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from FloatTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from FloatTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from FloatTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	Gets or sets a value that specifies how output values are calculated for the animation.
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from FloatTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from FloatTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from FloatTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from FloatTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from FloatTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from FloatTimedModifier.
KeyFrames	KeyValues
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from FloatTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from FloatTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from FloatTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself until the time specified by this property. Inherited from FloatTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from FloatTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from FloatTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the

	animation, compared to its parent timeline. Inherited from FloatTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from FloatTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
```

```
   Height="30" Width="200">

   <Button.Background>
     <SolidColorBrush Color="Blue">
       <SolidColorBrush.ColorAnimations>
         <ColorAnimation From="Red" To="Blue" Duration="7"
           RepeatCount="500" AutoReverse="True"/>
       </SolidColorBrush.ColorAnimations>
     </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.
For more information about animating properties, see Animation in "Avalon".

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "XAML", set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.

In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
        <Canvas ID="root"
  xmlns="http://schemas.microsoft.com/2003/xaml">

  <Line X1="10" Y1="20" X2="50" Y2="20"
    StrokeThickness="10" Stroke="Black">
    <Line.X2>
      <LengthAnimationCollection>
        <LengthAnimation From="30" To="300" Duration="10"
          RepeatDuration="Indefinite" />
      </LengthAnimationCollection>
    </Line.X2>
  </Line>

</Canvas>
```

```
        // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
```

```
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);

        ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

FloatAnimationCollection Class
Definition: Represents a collection of FloatModifier animations.

Method	Description
Add	The Add(FloatModifier) and Add(Object) methods add animations to the collection. The Add(Single,FloatAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified FloatModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this FloatAnimationCollection.
CopyTo	Copies the entire FloatAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.

EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FloatAnimationCollection	Initializes a new instance of the FloatAnimationCollection class.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates and returns the output of the animation collection.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the

	OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	Gets the type of animation stored in the collection.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty FloatAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	Gets or sets the animation at the specified index.
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as FloatAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of a FloatAnimationCollection, the property calls the FloatAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.
Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
       <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

  <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Width>
     <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
       AutoReverse="True"/>
     </LengthAnimationCollection>
   </Button.Width>

   A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Background>
     <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
         RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
     </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.
For more information about animating properties, see Animation in "Avalon".

FloatKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a new FloatKeyFrameCollection
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FloatKeyFrameCollection	
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from

	a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

FloatModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this FloatModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from

	Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FloatModifier	
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a FloatModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Pr perty	Descripti n
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

FloatTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this FloatModifier Inherited from FloatModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this FloatTimedModifier
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the

	ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FloatTimedModifier	
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from FloatModifier.
GetValueImpl	Inherited from FloatModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a FloatModifier Inherited from FloatModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.

WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it

	will repeat itself until the time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

IntAnimationCollection Class

Definition: Represents a collection of IntModifier animations.

Method	Description
Add	The Add(IntModifier) and Add(Object) methods add animations to the collection. The Add(Int32,IntAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified IntModifier.
Copy	Creates a copy of this IntAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	Copies the entire IntAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from

	Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
IntAnimationCollection	Creates an empty IntAnimationCollection with a default capacity for a single animation.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from

	Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty IntAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as IntAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an IntAnimationCollection, the property calls the IntAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
          RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>

  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

IntModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this IntModifier
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
IntModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.

OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a IntModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

IntTimedM difier Class

Method	Description

BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this IntTimedModifier
Copy	Creates a copy of this IntModifier Inherited from IntModifier.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from IntModifier.
GetValueImpl	Inherited from IntModifier.
IModifier.GetValue	Inherited from Modifier.
IntTimedModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.

OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a IntModifier Inherited from IntModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only

	used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself until the time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

LengthAnimation Class
Definition: Used to animate properties that accept a Length value.

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from LengthTimedModifier.

CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this LengthTimedModifier Inherited from LengthTimedModifier.
Copy	Creates a copy of this LengthModifier Inherited from LengthModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from LengthTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from LengthTimedModifier.
EndIn	Schedules an interactive end time. Inherited from LengthTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from LengthTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates the value of the animation at the current time.
GetValueImpl	Inherited from LengthModifier.
IModifier.GetValue	Inherited from Modifier.
LengthAnimation	Initializes a new instance of the LengthAnimation class.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.

op_Implicit	Implicitly creates an AnimationCollection from a LengthModifier Inherited from LengthModifier.
Pause	Pauses this timeline. Inherited from LengthTimedModifier.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from LengthTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from LengthTimedModifier.
SetDefaultParentTimeline	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from LengthTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from LengthTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from LengthTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from LengthTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from LengthTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from LengthTimedModifier.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an

	animation. Inherited from LengthTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from LengthTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from LengthTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from LengthTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from LengthTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	InterpolationMethod
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from LengthTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from LengthTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from LengthTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from LengthTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from LengthTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from LengthTimedModifier.
KeyFrames	KeyValues
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from LengthTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from LengthTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from LengthTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is great than the simple duration of the animation, it will repeat itself for the length of time specified by this property. Inherited from LengthTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from LengthTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from LengthTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from LengthTimedModifier.

StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from LengthTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to animate the size of an element using "Longhorn" markup language (code-named "XAML"). There are multiple ways to animate the size of an element: directly animate the height and width attributes of the element, or apply an animated ScaleTransform to the element. In this example, two Rectangle elements are resized using these methods. One rectangle is resized by animating its RectangleWidth attribute and another is resized by animating a ScaleTransform applied to the rectangle. Each rectangle is filled with a pattern to highlight the differences between the two resizing methods. Initially, the two patterns look the same, but as the rectangles are resized, patterns change depending on how their containing rectangle is resized.

In the first example, a Rectangle element's RectangleWidth property is animated using a LengthAnimationCollection and a LengthAnimation. The LengthAnimation object in this example animates the rectangle's RectangleWidth from its base value of 200 To a destination value of 600 over a Duration of 10 seconds.

```
      <Canvas ID="root"
xmlns="http://schemas.microsoft.com/2003/xaml">

<Rectangle
  RectangleTop="20"
  RectangleLeft="20"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Red"
  StrokeThickness="5">

  <Rectangle.Fill>
    <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
      ImageSource="help.gif" TileMode="Tile"/>
  </Rectangle.Fill>

  <Rectangle.RectangleWidth>
  <!-- Animate the Rectangle's width: -->
    <LengthAnimationCollection>
      <LengthAnimation
        To="600" Duration="10" AutoReverse="true" RepeatCount="50" />
    </LengthAnimationCollection>
  </Rectangle.RectangleWidth>

</Rectangle>
```

When the previous "XAML" is run, more of the pattern is exposed as the rectangle expands; however, the question marks that make up the pattern do not grow larger.

In the next example, a TransformDecorator is used to apply a ScaleTransform to a rectangle. A DoubleAnimation is used to animate the ScaleTransform object's ScaleX value using the ScaleXAnimations attribute. The DoubleAnimation animates the ScaleX value From 1 To a destination value of 3 over a Duration of 10 seconds. As a result, the rectangle's width is scaled from 100 percent (its starting size) to 300 percent over ten seconds.

```
 <TransformDecorator AffectsLayout="False">
  <TransformDecorator.Transform>

  <!-- Use the ScaleTransform to enlarge the rectangle -->
   <ScaleTransform ScaleX="1" ScaleY="1">
    <ScaleTransform.ScaleXAnimations>
     <DoubleAnimation  From="1" To="3" RepeatCount="30"
      AutoReverse="True" Begin="0" Duration="10" />
    </ScaleTransform.ScaleXAnimations>
   </ScaleTransform>

  </TransformDecorator.Transform>

  <Rectangle
   RectangleLeft="20"
   RectangleTop="200"
   RectangleWidth="200"
   RectangleHeight="150"
   Stroke="Black"
   StrokeThickness="3">
    <Rectangle.Fill>
     <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
      ImageSource="help.gif" TileMode="Tile"/>
     </Rectangle.Fill>
  </Rectangle>
 </TransformDecorator>

</Canvas>
```

In the previous example, the DoubleAnimationCollection tag, <DoubleAnimationCollection>, is omitted when animating the transformation's scale factor. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ScaleXAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

When the second rectangle expands, the objects in the pattern also grow larger, unlike in the first rectangle. The pattern behaves this way because when you transform an element the entire element and its child elements are transformed. When you directly alter the size of an element, as in the case of the first rectangle, the element's children are not resized, unless their size and position are dependent on the size of their parent element.

This example demonstrates how to use the By, From, and To properties of animations to set an animation's starting and ending values in "XAML". In the following markup, LengthAnimation objects are used to animate the endpoints of five Line elements. Although this example uses the LengthAnimation, the behavior of the From, To, and By properties is the same for all the animation classes.

In the first markup fragment, the X2 attribute of the first line is animated from 50 to 100 over a duration of 10 seconds. Because the From and To properties of the LengthAnimation are set, the animation ignores the line's base value, starting at the specified From value and moving toward the specified To value.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Line ID="Line1" X1="10" Y1="10" X2="100" Y2="50"
  Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" To="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The second line's animation has only its To property set. When the From value of an animation isn't set, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value.

```
        <Line ID="Line2" X1="10" Y1="70" X2="100" Y2="70" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation To="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The third line's animation has only its By property set. The By of an animation specifies "by how much" the animation changes a value over its duration. As in the previous example, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value, and adds 300 to that value over a duration of 10 seconds.

```
        <Line ID="Line3" X1="10" Y1="130" X2="100" Y2="130" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fourth line's animation has its By and From properties set. As a result, the line's X2 attribute is animated from 50 to to 350 over a duration of 10 seconds.

```
        <Line ID="Line4" X1="10" Y1="190" X2="100" Y2="190" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fifth line's animation has only its From value set. When an animation has no explicit destination value, it uses the base value of the property it is animating or the output of a previous animation as its destination value. In this case, the line's X2 attribute is animated from 50 to 100.

```
        <Line ID="Line5" X1="10" Y1="250" X2="100" Y2="250" Stroke="Black" StrokeThickness="5">
   <Line.X2>
    <LengthAnimationCollection>
     <LengthAnimation From="50" Duration="10" RepeatCount="50" />
    </LengthAnimationCollection>
   </Line.X2>
  </Line>

</Canvas>
```

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "XAML", set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.

In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
        <Canvas ID="root"
  xmlns="http://schemas.microsoft.com/2003/xaml">

  <Line X1="10" Y1="20" X2="50" Y2="20"
   StrokeThickness="10" Stroke="Black">
   <Line.X2>
    <LengthAnimationCollection>
     <LengthAnimation From="30" To="300" Duration="10"
       RepeatDuration="Indefinite" />
    </LengthAnimationCollection>
   </Line.X2>
  </Line>

</Canvas>
```

```
        // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);
```

```
        ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

LengthAnimationCollection Class
Definition: Represents a collection of LengthModifier animations.

Method	Description
Add	
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this LengthAnimationCollection.
CopyTo	
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in

	hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
LengthAnimationCollection	Creates an empty LengthAnimationCollection with a default capacity for a single animation.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from

	Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty LengthAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as LengthAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an LengthAnimationCollection, the property calls the LengthAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate the size of an element using "Longhorn" markup language (code-named "XAML"). There are multiple ways to animate the size of an element: directly animate the height and width attributes of the element, or apply an animated ScaleTransform to the element. In this example, two Rectangle elements are resized using these methods. One rectangle is resized by animating its RectangleWidth attribute and another is resized by animating a ScaleTransform applied to the rectangle. Each rectangle is filled with a pattern to highlight the differences between the two resizing methods. Initially, the two patterns look the same, but as the rectangles are resized, patterns change depending on how their containing rectangle is resized.

In the first example, a Rectangle element's RectangleWidth property is animated using a LengthAnimationCollection and a LengthAnimation. The LengthAnimation object in this example animates the rectangle's RectangleWidth from its base value of 200 To a destination value of 600 over a Duration of 10 seconds.

```
<Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Rectangle
  RectangleTop="20"
  RectangleLeft="20"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Red"
  StrokeThickness="5">

  <Rectangle.Fill>
   <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
    ImageSource="help.gif" TileMode="Tile"/>
  </Rectangle.Fill>

  <Rectangle.RectangleWidth>
  <!-- Animate the Rectangle's width: -->
   <LengthAnimationCollection>
    <LengthAnimation
     To="600" Duration="10" AutoReverse="true" RepeatCount="50" />
   </LengthAnimationCollection>
  </Rectangle.RectangleWidth>

 </Rectangle>
```

When the previous "XAML" is run, more of the pattern is exposed as the rectangle expands; however, the question marks that make up the pattern do not grow larger.

In the next example, a TransformDecorator is used to apply a ScaleTransform to a rectangle. A DoubleAnimation is used to animate the ScaleTransform object's ScaleX value using the ScaleXAnimations attribute. The DoubleAnimation animates the ScaleX value From 1 To a destination value of 3 over a Duration of 10 seconds. As a result, the rectangle's width is scaled from 100 percent (its starting size) to 300 percent over ten seconds.

```
<TransformDecorator AffectsLayout="False">
 <TransformDecorator.Transform>

 <!-- Use the ScaleTransform to enlarge the rectangle -->
  <ScaleTransform ScaleX="1" ScaleY="1">
   <ScaleTransform.ScaleXAnimations>
    <DoubleAnimation  From="1" To="3" RepeatCount="30"
     AutoReverse="True" Begin="0" Duration="10" />
   </ScaleTransform.ScaleXAnimations>
  </ScaleTransform>

 </TransformDecorator.Transform>

 <Rectangle
  RectangleLeft="20"
  RectangleTop="200"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Black"
  StrokeThickness="3">
```

```
      <Rectangle.Fill>
        <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
          ImageSource="help.gif" TileMode="Tile"/>
        </Rectangle.Fill>
    </Rectangle>
  </TransformDecorator>

</Canvas>
```

In the previous example, the DoubleAnimationCollection tag, <DoubleAnimationCollection>, is omitted when animating the transformation's scale factor. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ScaleXAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

When the second rectangle expands, the objects in the pattern also grow larger, unlike in the first rectangle. The pattern behaves this way because when you transform an element the entire element and its child elements are transformed. When you directly alter the size of an element, as in the case of the first rectangle, the element's children are not resized, unless their size and position are dependent on the size of their parent element.

LengthKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a new LengthKeyFrameCollection
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
LengthKeyFrameCollection	

MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

LengthModifier Class

Meth d	Descripti n
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this LengthModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
LengthModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a LengthModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.

ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

This example demonstrates how to animate the size of an element using "Longhorn" markup language (code-named "XAML"). There are multiple ways to animate the size of an element: directly animate the height and width attributes of the element, or apply an animated ScaleTransform to the element. In this example, two Rectangle elements are resized using these methods. One rectangle is resized by animating its RectangleWidth attribute and another is resized by animating a ScaleTransform applied to the rectangle. Each rectangle is filled with a pattern to highlight the differences between the two resizing methods. Initially, the two patterns look the same, but as the rectangles are resized, patterns change depending on how their containing rectangle is resized.

In the first example, a Rectangle element's RectangleWidth property is animated using a LengthAnimationCollection and a LengthAnimation. The LengthAnimation object in this example animates the rectangle's RectangleWidth from its base value of 200 To a destination value of 600 over a Duration of 10 seconds.

```
<Canvas ID="root"
```

```
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Rectangle
  RectangleTop="20"
  RectangleLeft="20"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Red"
  StrokeThickness="5">

  <Rectangle.Fill>
   <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
    ImageSource="help.gif' TileMode="Tile"/>
  </Rectangle.Fill>

  <Rectangle.RectangleWidth>
  <!-- Animate the Rectangle's width: -->
   <LengthAnimationCollection>
    <LengthAnimation
     To="600" Duration="10" AutoReverse="true" RepeatCount="50" />
   </LengthAnimationCollection>
  </Rectangle.RectangleWidth>

</Rectangle>
```

When the previous "XAML" is run, more of the pattern is exposed as the rectangle expands; however, the question marks that make up the pattern do not grow larger.

In the next example, a TransformDecorator is used to apply a ScaleTransform to a rectangle. A DoubleAnimation is used to animate the ScaleTransform object's ScaleX value using the ScaleXAnimations attribute. The DoubleAnimation animates the ScaleX value From 1 To a destination value of 3 over a Duration of 10 seconds. As a result, the rectangle's width is scaled from 100 percent (its starting size) to 300 percent over ten seconds.

```
<TransformDecorator AffectsLayout="False">
 <TransformDecorator.Transform>

 <!-- Use the ScaleTransform to enlarge the rectangle -->
  <ScaleTransform ScaleX="1" ScaleY="1">
   <ScaleTransform.ScaleXAnimations>
    <DoubleAnimation  From="1" To="3" RepeatCount="30"
     AutoReverse="True" Begin="0" Duration="10" />
   </ScaleTransform.ScaleXAnimations>
  </ScaleTransform>

 </TransformDecorator.Transform>

 <Rectangle
  RectangleLeft="20"
  RectangleTop="200"
  RectangleWidth="200"
  RectangleHeight="150"
  Stroke="Black"
  StrokeThickness="3">
   <Rectangle.Fill>
    <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
```

```
        ImageSource="help.gif" TileMode="Tile"/>
      </Rectangle.Fill>
   </Rectangle>
  </TransformDecorator>

</Canvas>
```

In the previous example, the DoubleAnimationCollection tag, <DoubleAnimationCollection>, is omitted when animating the transformation's scale factor. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ScaleXAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

When the second rectangle expands, the objects in the pattern also grow larger, unlike in the first rectangle. The pattern behaves this way because when you transform an element the entire element and its child elements are transformed. When you directly alter the size of an element, as in the case of the first rectangle, the element's children are not resized, unless their size and position are dependent on the size of their parent element.

LengthTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this LengthModifier Inherited from LengthModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this LengthTimedModifier
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	

Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from LengthModifier.
GetValueImpl	Inherited from LengthModifier.
IModifier.GetValue	Inherited from Modifier.
LengthTimedModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a LengthModifier Inherited from LengthModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Descripti n
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.

AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is great than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the

	animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

This example demonstrates how to animate the size of an element using "Longhorn" markup language (code-named "XAML"). There are multiple ways to animate the size of an element: directly animate the height and width attributes of the element, or apply an animated ScaleTransform to the element. In this example, two Rectangle elements are resized using these methods. One rectangle is resized by animating its RectangleWidth attribute and another is resized by animating a ScaleTransform applied to the rectangle. Each rectangle is filled with a pattern to highlight the differences between the two resizing methods. Initially, the two patterns look the same, but as the rectangles are resized, patterns change depending on how their containing rectangle is resized.

In the first example, a Rectangle element's RectangleWidth property is animated using a LengthAnimationCollection and a LengthAnimation. The LengthAnimation object in this example animates the rectangle's RectangleWidth from its base value of 200 To a destination value of 600 over a Duration of 10 seconds.

```
<Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

  <Rectangle
    RectangleTop="20"
    RectangleLeft="20"
    RectangleWidth="200"
    RectangleHeight="150"
    Stroke="Red"
    StrokeThickness="5">

    <Rectangle.Fill>
      <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
        ImageSource="help.gif" TileMode="Tile"/>
    </Rectangle.Fill>

    <Rectangle.RectangleWidth>
    <!-- Animate the Rectangle's width: -->
      <LengthAnimationCollection>
        <LengthAnimation
          To="600" Duration="10" AutoReverse="true" RepeatCount="50" />
      </LengthAnimationCollection>
    </Rectangle.RectangleWidth>

  </Rectangle>
```

When the previous "XAML" is run, more of the pattern is exposed as the rectangle expands; however, the question marks that make up the pattern do not grow larger.

In the next example, a TransformDecorator is used to apply a ScaleTransform to a rectangle. A DoubleAnimation is used to animate the ScaleTransform object's ScaleX value using the ScaleXAnimations attribute. The DoubleAnimation animates the ScaleX value From 1 To a destination value of 3 over a Duration of 10 seconds. As a result, the rectangle's width is scaled from 100 percent (its starting size) to 300 percent over ten seconds.

```
  <TransformDecorator AffectsLayout="False">
   <TransformDecorator.Transform>

   <!-- Use the ScaleTransform to enlarge the rectangle -->
    <ScaleTransform ScaleX="1" ScaleY="1">
     <ScaleTransform.ScaleXAnimations>
      <DoubleAnimation  From="1" To="3" RepeatCount="30"
       AutoReverse="True" Begin="0" Duration="10" />
     </ScaleTransform.ScaleXAnimations>
    </ScaleTransform>

   </TransformDecorator.Transform>

   <Rectangle
    RectangleLeft="20"
    RectangleTop="200"
    RectangleWidth="200"
    RectangleHeight="150"
    Stroke="Black"
    StrokeThickness="3">
     <Rectangle.Fill>
      <ImageBrush ViewPort="0,0 100,100" ViewPortUnits="Absolute"
       ImageSource="help.gif" TileMode="Tile"/>
      </Rectangle.Fill>
   </Rectangle>
  </TransformDecorator>

</Canvas>
```

In the previous example, the DoubleAnimationCollection tag, <DoubleAnimationCollection>, is omitted when animating the transformation's scale factor. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ScaleXAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

When the second rectangle expands, the objects in the pattern also grow larger, unlike in the first rectangle. The pattern behaves this way because when you transform an element the entire element and its child elements are transformed. When you directly alter the size of an element, as in the case of the first rectangle, the element's children are not resized, unless their size and position are dependent on the size of their parent element.

L ngAnimationC llection Class
Definition: Represents a collection of LongModifier animations.

Meth d	Descripti n
Add	The Add(LongModifier) and Add(Object) methods add animations to the collection. The Add(Int64,LongAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified LongModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this LongAnimationCollection.
CopyTo	Copies the entire LongAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	

LongAnimationCollection	Creates an empty LongAnimationCollection with a default capacity for a single animation.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty LongAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.

IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as LongAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an LongAnimationCollection, the property calls the LongAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
```

```
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
      <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
          RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>

  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

LongModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this LongModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.

EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
LongModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a LongModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed

	from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

LongTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this LongModifier Inherited from LongModifier.
Copy	Creates a copy of this LongTimedModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a

EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from LongModifier.
GetValueImpl	Inherited from LongModifier.
IModifier.GetValue	Inherited from Modifier.
LongTimedModifier	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a LongModifier Inherited from LongModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from

	Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.

RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

MatrixAnimationCollection Class
Definition: Represents a collection of MatrixModifier animations.

Method	Description
Add	
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	
Copy	Creates a copy of this MatrixAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be

EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MatrixAnimationCollection	Creates an empty MatrixAnimationCollection with a default capacity for a single animation.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.

ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty MatrixAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as MatrixAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an MatrixAnimationCollection, the property calls the MatrixAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
          RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>
  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

MatrixModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this MatrixModifier
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MatrixModifier	
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this

	method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a MatrixModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

MatrixTimedModifier Class

Meth d	Description

BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this MatrixModifier Inherited from MatrixModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this MatrixTimedModifier
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from MatrixModifier.
GetValueImpl	Inherited from MatrixModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MatrixTimedModifier	
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from

	Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a MatrixModifier Inherited from MatrixModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.

Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

Modifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.

Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValueImpl	Used to be internal. Can be private or something when we move to generics.
IModifier.GetValue	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
Modifier	
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.

WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Pr perty	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue

ObjectAnimationCollection Class
Definition: Represents a collection of ObjectModifier animations.

Method	Description
Add	The Add(ObjectModifier) and Add(Object) methods add animations to the collection. The Add(Object,ObjectAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified ObjectModifier.
Copy	Creates a copy of this ObjectAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.

CopyTo	Copies the entire ObjectAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
ObjectAnimationCollection	Creates an empty ObjectAnimationCollection with a default capacity for a single animation.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.

Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty ObjectAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for

maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as ObjectAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an ObjectAnimationCollection, the property calls the ObjectAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
      <ColorAnimation From="Red" To="Blue" Duration="7"
        RepeatCount="500" AutoReverse="True"/>
     </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

ObjectModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this ObjectModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.

MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
ObjectModifier	
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a ObjectModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

UsesBaseValue	UsesBaseValue Inherited from Modifier.

ObjectTimedM difier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this ObjectTimedModifier
Copy	Creates a copy of this ObjectModifier Inherited from ObjectModifier.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from ObjectModifier.
GetValueImpl	Inherited from ObjectModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified

	object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
ObjectTimedModifier	
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a ObjectModifier Inherited from ObjectModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the

	implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is great than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

PathAnimation Class
Definition: This animation can be used inside of a MatrixAnimationCollection to move a visual object along a path.

Method	Descripti n
BeginIn	Starts or restarts the animation at the specified offset from the current

	time. Inherited from MatrixTimedModifier.
CloneCore	
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this MatrixTimedModifier Inherited from MatrixTimedModifier.
Copy	Creates a copy of this MatrixModifier Inherited from MatrixModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from MatrixTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from MatrixTimedModifier.
EndIn	Schedules an interactive end time. Inherited from MatrixTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from MatrixTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates the current matrix value for this animation.
GetValueImpl	Inherited from MatrixModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.

op_Implicit	Implicitly creates an AnimationCollection from a MatrixModifier Inherited from MatrixModifier.
PathAnimation	Creates a new PathAnimation class.
Pause	Pauses this timeline. Inherited from MatrixTimedModifier.
PropagateEventHandler	Inherited from MatrixTimedModifier.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from MatrixTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from MatrixTimedModifier.
SetDefaultParentTimeline	Changes the default parent timeline for this animation.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from MatrixTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from MatrixTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from MatrixTimedModifier.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from MatrixTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from MatrixTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from MatrixTimedModifier.
DoesRotateWithTangent	If this is set to true, the object will rotate along with the tangent to the path.

Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from MatrixTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from MatrixTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from MatrixTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from MatrixTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from MatrixTimedModifier.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from MatrixTimedModifier.
IsEnabled	Inherited from MatrixTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from MatrixTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from MatrixTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from MatrixTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from MatrixTimedModifier.
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from MatrixTimedModifier.
PathGeometry	This geometry specifies the path.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from MatrixTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from MatrixTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property. Inherited from MatrixTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from MatrixTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from MatrixTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from MatrixTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object

	is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from MatrixTimedModifier.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	Currently always returns false.

PointAnimation Class
Definition: Used to animate properties that accept Point values.

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from PointTimedModifier.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this PointModifier Inherited from PointModifier.
Copy	Creates a copy of this PointTimedModifier Inherited from PointTimedModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from PointTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from PointTimedModifier.
EndIn	Schedules an interactive end time. Inherited from PointTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from PointTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.

GetValue	Calculates the value of the animation at the current time.
GetValueImpl	Inherited from PointModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a PointModifier Inherited from PointModifier.
Pause	Pauses this timeline. Inherited from PointTimedModifier.
PointAnimation	Initializes a new instance of the PointAnimation class.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from PointTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from PointTimedModifier.
SetDefaultParentTimeline	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from PointTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from

	PointTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from PointTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from PointTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from PointTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from PointTimedModifier.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from PointTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from PointTimedModifier.
EndSync	Not supported. Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from PointTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from PointTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from PointTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	Gets or sets a value that specifies how output values are calculated for the animation.
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from PointTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from PointTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from PointTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from PointTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from PointTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from PointTimedModifier.
KeyFrames	KeyValues
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited

	from PointTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from PointTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from PointTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property. Inherited from PointTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from PointTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from PointTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from PointTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from PointTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to use the By, From, and To properties of animations to set an animation's starting and ending values in "Longhorn" markup language (code-named "XAML"). In the following markup, LengthAnimation objects are used to animate the endpoints of five Line elements. Although this example uses the LengthAnimation, the behavior of the From, To, and By properties is the same for all the animation classes.

In the first markup fragment, the X2 attribute of the first line is animated from 50 to 100 over a duration of 10 seconds. Because the From and To properties of the LengthAnimation are set, the animation ignores the line's base value, starting at the specified From value and moving toward the specified To value.

```
      <Canvas ID="root"
xmlns="http://schemas.microsoft.com/2003/xaml">

<Line ID="Line1" X1="10" Y1="10" X2="100" Y2="50"
 Stroke="Black" StrokeThickness="5">
 <Line.X2>
  <LengthAnimationCollection>
   <LengthAnimation From="50" To="300" Duration="10" RepeatCount="50" />
  </LengthAnimationCollection>
 </Line.X2>
</Line>
```

The second line's animation has only its To property set. When the From value of an animation isn't set, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value.

```
      <Line ID="Line2" X1="10" Y1="70" X2="100" Y2="70" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation To="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The third line's animation has only its By property set. The By of an animation specifies "by how much" the animation changes a value over its duration. As in the previous example, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value, and adds 300 to that value over a duration of 10 seconds.

```
      <Line ID="Line3" X1="10" Y1="130" X2="100" Y2="130" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fourth line's animation has its By and From properties set. As a result, the line's X2 attribute is animated from 50 to to 350 over a duration of 10 seconds.

```
      <Line ID="Line4" X1="10" Y1="190" X2="100" Y2="190" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fifth line's animation has only its From value set. When an animation has no explicit destination value, it uses the base value of the property it is animating or the output of a previous animation as its destination value. In this case, the line's X2 attribute is animated from 50 to 100.

```
      <Line ID="Line5" X1="10" Y1="250" X2="100" Y2="250" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>

</Canvas>
```

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "XAML", set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.

In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Line X1="10" Y1="20" X2="50" Y2="20"
   StrokeThickness="10" Stroke="Black">
   <Line.X2>
     <LengthAnimationCollection>
       <LengthAnimation From="30" To="300" Duration="10"
         RepeatDuration="Indefinite" />
     </LengthAnimationCollection>
   </Line.X2>
 </Line>

</Canvas>
```

```
        // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);
```

```
        ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

PointAnimationCollection Class
Definition: Represents a collection of PointModifier animations.

Method	Descripti n
Add	The Add(PointModifier) and Add(Object) methods add animations to the collection. The Add(Point,PointAnimationCollection) method calculates

	the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified PointModifier.
Copy	Creates a copy of this PointAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	Copies the entire PointAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates and returns the output of the animation collection.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.

MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PointAnimationCollection	Initializes a new instance of the PointAnimationCollection class.
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	Gets the type of animation stored in the collection. PointAnimationCollection objects always return the type of PointAnimation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty PointAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	

IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	Gets or sets the animation at the specified index.
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as PointAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of a PointAnimationCollection, the property calls the PointAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
      <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
     <LengthAnimation To="50" Duration="5" RepeatCount="500"
      AutoReverse="True"/>
    </LengthAnimationCollection>
   </Button.Width>

   A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Background>
    <SolidColorBrush Color="Blue">
     <SolidColorBrush.ColorAnimations>
      <ColorAnimation From="Red" To="Blue" Duration="7"
        RepeatCount="500" AutoReverse="True"/>
     </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

PointKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a new PointKeyFrameCollection
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.

EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PointKeyFrameCollection	
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.

Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

PointModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this PointModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.

MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit PointModifier	Implicitly creates an AnimationCollection from a PointModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

UsesBaseValue	UsesBaseValue Inherited from Modifier.

PointTimedM difier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this PointTimedModifier
Copy	Creates a copy of this PointModifier Inherited from PointModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from PointModifier.
GetValueImpl	Inherited from PointModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.

ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a PointModifier Inherited from PointModifier.
Pause	Pauses this timeline.
PointTimedModifier	
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a

	single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

RectAnimation Class
Definition: Used to animate properties that accept a Rect value.

Meth d	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from RectTimedModifier.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this RectTimedModifier Inherited from RectTimedModifier.
Copy	Creates a copy of this RectModifier Inherited from RectModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from RectTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from RectTimedModifier.
EndIn	Schedules an interactive end time. Inherited from RectTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from RectTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates the value of the animation at the current time.
GetValueImpl	Inherited from RectModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this

	method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a RectModifier Inherited from RectModifier.
Pause	Pauses this timeline. Inherited from RectTimedModifier.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
RectAnimation	Creates a new RectAnimation with all properties set to their default values.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from RectTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from RectTimedModifier.
SetDefaultParentTimeline	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from RectTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from RectTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from RectTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from RectTimedModifier.

CurrentTime	Gets the current time value of the animation. Inherited from RectTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from RectTimedModifier.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from RectTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from RectTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from RectTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from RectTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from RectTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	InterpolationMethod
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from RectTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from RectTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from RectTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from RectTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from RectTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from RectTimedModifier.
KeyFrames	KeyValues
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from RectTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from RectTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from RectTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property. Inherited from RectTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is

	reached. Inherited from RectTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from RectTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from RectTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from RectTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to use the By, From, and To properties of animations to set an animation's starting and ending values in "Longhorn" markup language (code-named "XAML"). In the following markup, LengthAnimation objects are used to animate the endpoints of five Line elements. Although this example uses the LengthAnimation, the behavior of the From, To, and By properties is the same for all the animation classes.

In the first markup fragment, the X2 attribute of the first line is animated from 50 to 100 over a duration of 10 seconds. Because the From and To properties of the LengthAnimation are set, the animation ignores the line's base value, starting at the specified From value and moving toward the specified To value.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Line ID="Line1" X1="10" Y1="10" X2="100" Y2="50"
   Stroke="Black" StrokeThickness="5">
   <Line.X2>
     <LengthAnimationCollection>
       <LengthAnimation From="50" To="300" Duration="10" RepeatCount="50" />
     </LengthAnimationCollection>
   </Line.X2>
 </Line>
```

The second line's animation has only its To property set. When the From value of an animation isn't set, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value.

```
        <Line ID="Line2" X1="10" Y1="70" X2="100" Y2="70" Stroke="Black" StrokeThickness="5">
   <Line.X2>
     <LengthAnimationCollection>
       <LengthAnimation To="300" Duration="10" RepeatCount="50" />
     </LengthAnimationCollection>
   </Line.X2>
 </Line>
```

The third line's animation has only its By property set. The By of an animation specifies "by how much" the animation changes a value over its duration. As in the previous example, the animation uses the base value of the property it is animating or the ending value of a previous animation. In this example, the animation uses the base value of the line's X2 property, 100, as its starting value, and adds 300 to that value over a duration of 10 seconds.

```
      <Line ID="Line3" X1="10" Y1="130" X2="100" Y2="130" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fourth line's animation has its By and From properties set. As a result, the line's X2 attribute is animated from 50 to to 350 over a duration of 10 seconds.

```
      <Line ID="Line4" X1="10" Y1="190" X2="100" Y2="190" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" By="300" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>
```

The fifth line's animation has only its From value set. When an animation has no explicit destination value, it uses the base value of the property it is animating or the output of a previous animation as its destination value. In this case, the line's X2 attribute is animated from 50 to 100.

```
      <Line ID="Line5" X1="10" Y1="250" X2="100" Y2="250" Stroke="Black" StrokeThickness="5">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="50" Duration="10" RepeatCount="50" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>

</Canvas>
```

This example demonstrates how to animate a property using "XAML". To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
      <Canvas ID="root"
```

```
  xmlns="http://schemas.microsoft.com/2003/xaml">

  <Button Canvas.Top="20" Canvas.Left="20"
    Height="30" Width="200">

   <Button.Width>
      <LengthAnimationCollection>
        <LengthAnimation To="50" Duration="5" RepeatCount="500"
          AutoReverse="True"/>
      </LengthAnimationCollection>
    </Button.Width>

    A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
            <Button Canvas.Top="70" Canvas.Left="20"
    Height="30" Width="200">

    <Button.Background>
      <SolidColorBrush Color="Blue">
        <SolidColorBrush.ColorAnimations>
          <ColorAnimation From="Red" To="Blue" Duration="7"
            RepeatCount="500" AutoReverse="True"/>
        </SolidColorBrush.ColorAnimations>
      </SolidColorBrush>
    </Button.Background>

    Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.
For more information about animating properties, see Animation in "Avalon".

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "XAML", set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.
In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
            <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">
```

```
  <Line X1="10" Y1="20" X2="50" Y2="20"
   StrokeThickness="10" Stroke="Black">
   <Line.X2>
    <LengthAnimationCollection>
     <LengthAnimation From="30" To="300" Duration="10"
      RepeatDuration="Indefinite" />
    </LengthAnimationCollection>
   </Line.X2>
  </Line>

</Canvas>
```

```
      // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);
```

```
      ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
 MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

RectAnimationCollection Class
Definition: Represents a collection of RectModifier animations.

Method	Description
Add	The Add(RectModifier) and Add(Object) methods add animations to the collection. The Add(Rect,RectAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.

Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified RectModifier.
Copy	Creates a copy of this RectAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	Copies the entire RectAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this

	method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
RectAnimationCollection	Creates an empty RectAnimationCollection with a default capacity for a single animation.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty RectAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.

IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as RectAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an RectAnimationCollection, the property calls the RectAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
      <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
          RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>

  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

RectKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a new RectKeyFrameCollection
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup

	operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
RectKeyFrameCollection	
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	

StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

RectModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this RectModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.

OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a RectModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
RectModifier	
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

RectTimedModifier Class

Meth d	Descripti n
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this RectTimedModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this RectModifier Inherited from RectModifier.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from RectModifier.
GetValueImpl	Inherited from RectModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from

	Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a RectModifier Inherited from RectModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
RectTimedModifier	
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only

	used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

ShortAnimationCollection Class

Definition: Represents a collection of ShortModifier animations.

Method	Descripti n
Add	The Add(ShortModifier) and Add(Object) methods add animations to the collection; the Add(Int16,ShortAnimationCollection) method calculates the current value of the specified collection based on the specified base

	value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified ShortModifier.
Copy	Creates a copy of this ShortAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	Copies the entire ShortAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.

ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ShortAnimationCollection	Creates an empty ShortAnimationCollection with a default capacity for a single animation.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty ShortAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.

IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as ShortAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an ShortAnimationCollection, the property calls the ShortAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

ShortModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this ShortModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.

GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a ShortModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ShortModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.

IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

ShortTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this ShortModifier Inherited from ShortModifier.
Copy	Creates a copy of this ShortTimedModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in

	hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from ShortModifier.
GetValueImpl	Inherited from ShortModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a ShortModifier Inherited from ShortModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ShortTimedModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.

AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a

	Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

SizeAnimation Class

Definition: Defines an animation based on the Size of an object. By providing Size information, an object can appear to shrink or enlarge over a period of time.

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from SizeTimedModifier.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this SizeTimedModifier Inherited from SizeTimedModifier.
Copy	Creates a copy of this SizeModifier Inherited from SizeModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from SizeTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from SizeTimedModifier.
EndIn	Schedules an interactive end time. Inherited from SizeTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from SizeTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.

GetValue	Calculates the current value of the animation from the specified base value.
GetValueImpl	Inherited from SizeModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a SizeModifier Inherited from SizeModifier.
Pause	Pauses this timeline. Inherited from SizeTimedModifier.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from SizeTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from SizeTimedModifier.
SetDefaultParentTimeline	
SizeAnimation	Initializes a new instance of the SizeAnimation class.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from SizeTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.

AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from SizeTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from SizeTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from SizeTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from SizeTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from SizeTimedModifier.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from SizeTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from SizeTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from SizeTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from SizeTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from SizeTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	Gets or sets a value that specifies how output values are calculated for the animation.
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from SizeTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from SizeTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from SizeTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from SizeTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from SizeTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from SizeTimedModifier.
KeyFrames	KeyValues

ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from SizeTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from SizeTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from SizeTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is great than the simple duration of the animation, it will repeat itself until the time specified by this property. Inherited from SizeTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from SizeTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from SizeTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from SizeTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from SizeTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "Longhorn" markup language (code-named "XAML"), set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.

In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Line X1="10" Y1="20" X2="50" Y2="20"
  StrokeThickness="10" Stroke="Black">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="30" To="300" Duration="10"
     RepeatDuration="Indefinite" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>

</Canvas>
```

```
    // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);

    ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

The previous examples use classes from the MSAvalon.Windows.Media.Animation, MSAvalon.Windows, and MSAvalon.Windows.Shapes namespaces.

This example demonstrates how to animate a property in code. To animate a property, you associate the proper animation collection and animations with the property, either directly or using the property's corresponding animation property. There are a variety of animation classes in the MSAvalon.Windows.Media.Animation namespace, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the Rectangle must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. After the animation is created, it is applied to the button's Width property using the SetAnimations method. When using the SetAnimations property, you pass it the static property identifier field (in this example Button.WidthProperty) for the class and the AnimationCollection containing the animations. The code used to set the button's size and position has been omitted.

```
    // C#
Button aButton = new Button();

// Animate the Button's Width.
LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.To = new Length(50);
myLengthAnimation.Duration = new Time(5000);
myLengthAnimation.AutoReverse = true;
myLengthAnimation.RepeatDuration = Time.Indefinite;
LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

// Set the animation.
aButton.SetAnimations(Button.WidthProperty, collection);

        ' VB .NET
Dim aButton As MSAvalon.Windows.Controls.Button
aButton = new MSAvalon.Windows.Controls.Button
MSAvalon.Windows.Controls.Canvas.SetLeft(aButton, new MSAvalon.Windows.Length(20))
MSAvalon.Windows.Controls.Canvas.SetTop(aButton, new MSAvalon.Windows.Length(20))
aButton.Width = new MSAvalon.Windows.Length(200)
aButton.Height = new MSAvalon.Windows.Length(30)
aButton.Content = "A Button"

' Animate the Button's Width.
Dim myLengthAnimation As MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation = new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.To = new MSAvalon.Windows.Length(50)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(5000)
myLengthAnimation.AutoReverse = true
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite
Dim collection As MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection = new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

aButton.SetAnimations(MSAvalon.Windows.Controls.Button.WidthProperty, _
  collection)
```

In the previous example, the starting value of the animation isn't specified, so the animation uses the value of the button's Width as its starting value.

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimationCollection and a ColorAnimation are used to animate the property. The Color property has a corresponding ColorAnimations property, so the ColorAnimation is added to the ColorAnimations property's ColorAnimationCollection using the Add method.

```
        // C#
//Create and set the second Button.
Button anotherButton = new Button();

// Create and animate a Brush to set the Button's
// background.
SolidColorBrush myBrush = new SolidColorBrush();
myBrush.Color = Colors.Blue;

ColorAnimation myColorAnimation = new ColorAnimation();
myColorAnimation.From = Colors.Blue;
myColorAnimation.To = Colors.Red;
myColorAnimation.Duration = new Time(7000);
myColorAnimation.AutoReverse = true;
myColorAnimation.RepeatDuration = Time.Indefinite;

myBrush.ColorAnimations.Add(myColorAnimation);
anotherButton.Background = myBrush;


        ' VB .NET
' Create and set the second Button.
Dim anotherButton As MSAvalon.Windows.Controls.Button
anotherButton = new MSAvalon.Windows.Controls.Button
Canvas.SetLeft(anotherButton, new MSAvalon.Windows.Length(20))
Canvas.SetTop(anotherButton, new MSAvalon.Windows.Length(70))
anotherButton.Width = new MSAvalon.Windows.Length(200)
anotherButton.Height = new MSAvalon.Windows.Length(30)
anotherButton.Content = "Another Button"

' Create and animate a Brush to set the Button's fill.
Dim myBrush As MSAvalon.Windows.Media.SolidColorBrush
myBrush = new MSAvalon.Windows.Media.SolidColorBrush
myBrush.Color = MSAvalon.Windows.Media.Colors.Blue
Dim myColorAnimation As MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation = new MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation.From = MSAvalon.Windows.Media.Colors.Blue
myColorAnimation.To = MSAvalon.Windows.Media.Colors.Red
myColorAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(7000)
myColorAnimation.AutoReverse = true
myColorAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite
myBrush.ColorAnimations.Add(myColorAnimation)
anotherButton.Background = myBrush
```

This example uses the MSAvalon.Windows, MSAvalon.Windows.Controls, MSAvalon.Windows.Media, and MSAvalon.Windows.Media.Animation namespaces. For more information about animating properties, see Animation in "Avalon".

This example demonstrates how to animate a property using "XAML". To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">
```

```
  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
      <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
      <SolidColorBrush.ColorAnimations>
        <ColorAnimation From="Red" To="Blue" Duration="7"
          RepeatCount="500" AutoReverse="True"/>
      </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>

  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

SizeAnimationCollection Class

Definition: Represents a collection of SizeModifier animations.

Method	Description
Add	The Add(SizeModifier) and Add(Object) methods add animations to the collection; the Add(Size,SizeAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.

Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified SizeModifier.
Copy	Creates a copy of this SizeAnimationCollection.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
CopyTo	Copies the entire SizeAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates and returns the output of the animation collection.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.

ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
SizeAnimationCollection	Initializes a new instance of the SizeAnimationCollection class.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	Gets the type of animation stored in the collection. SizeAnimationCollection objects always return the type of SizeAnimation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty SizeAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.

IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	Gets or sets the animation at the specified index.
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as SizeAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of a SizeAnimationCollection, the property calls the SizeAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
        AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>
```

```
   A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
       <Button Canvas.Top="70" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Background>
     <SolidColorBrush Color="Blue">
       <SolidColorBrush.ColorAnimations>
         <ColorAnimation From="Red" To="Blue" Duration="7"
           RepeatCount="500" AutoReverse="True"/>
       </SolidColorBrush.ColorAnimations>
     </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.
For more information about animating properties, see Animation in "Avalon".

This example demonstrates how to animate a property in code. To animate a property, you associate the proper animation collection and animations with the property, either directly or using the property's corresponding animation property. There are a variety of animation classes in the MSAvalon.Windows.Media.Animation namespace, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the Rectangle must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. After the animation is created, it is applied to the button's Width property using the SetAnimations method. When using the SetAnimations property, you pass it the static property identifier field (in this example Button.WidthProperty) for the class and the AnimationCollection containing the animations. The code used to set the button's size and position has been omitted.

```
       // C#
Button aButton = new Button();

// Animate the Button's Width.
```

```
LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.To = new Length(50);
myLengthAnimation.Duration = new Time(5000);
myLengthAnimation.AutoReverse = true;
myLengthAnimation.RepeatDuration = Time.Indefinite;
LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

// Set the animation.
aButton.SetAnimations(Button.WidthProperty, collection);

        ' VB .NET
Dim aButton As MSAvalon.Windows.Controls.Button
aButton = new MSAvalon.Windows.Controls.Button
MSAvalon.Windows.Controls.Canvas.SetLeft(aButton, new MSAvalon.Windows.Length(20))
MSAvalon.Windows.Controls.Canvas.SetTop(aButton, new MSAvalon.Windows.Length(20))
aButton.Width = new MSAvalon.Windows.Length(200)
aButton.Height = new MSAvalon.Windows.Length(30)
aButton.Content = "A Button"

' Animate the Button's Width.
Dim myLengthAnimation As MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation = new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.To = new MSAvalon.Windows.Length(50)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(5000)
myLengthAnimation.AutoReverse = true
myLengthAnimation.RepeatDuration = _
 MSAvalon.Windows.Media.Animation.Time.Indefinite
Dim collection As MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection = new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

aButton.SetAnimations(MSAvalon.Windows.Controls.Button.WidthProperty, _
 collection)
```

In the previous example, the starting value of the animation isn't specified, so the animation uses the value of the button's Width as its starting value.

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimationCollection and a ColorAnimation are used to animate the property. The Color property has a corresponding ColorAnimations property, so the ColorAnimation is added to the ColorAnimations property's ColorAnimationCollection using the Add method.

```
        // C#
//Create and set the second Button.
Button anotherButton = new Button();

// Create and animate a Brush to set the Button's
// background.
SolidColorBrush myBrush = new SolidColorBrush();
myBrush.Color = Colors.Blue;

ColorAnimation myColorAnimation = new ColorAnimation();
```

```
myColorAnimation.From = Colors.Blue;
myColorAnimation.To = Colors.Red;
myColorAnimation.Duration = new Time(7000);
myColorAnimation.AutoReverse = true;
myColorAnimation.RepeatDuration = Time.Indefinite;

myBrush.ColorAnimations.Add(myColorAnimation);
anotherButton.Background = myBrush;
```

```
        ' VB .NET
' Create and set the second Button.
Dim anotherButton As MSAvalon.Windows.Controls.Button
anotherButton = new MSAvalon.Windows.Controls.Button
Canvas.SetLeft(anotherButton, new MSAvalon.Windows.Length(20))
Canvas.SetTop(anotherButton, new MSAvalon.Windows.Length(70))
anotherButton.Width = new MSAvalon.Windows.Length(200)
anotherButton.Height = new MSAvalon.Windows.Length(30)
anotherButton.Content = "Another Button"

' Create and animate a Brush to set the Button's fill.
Dim myBrush As MSAvalon.Windows.Media.SolidColorBrush
myBrush = new MSAvalon.Windows.Media.SolidColorBrush
myBrush.Color = MSAvalon.Windows.Media.Colors.Blue
Dim myColorAnimation As MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation = new MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation.From = MSAvalon.Windows.Media.Colors.Blue
myColorAnimation.To = MSAvalon.Windows.Media.Colors.Red
myColorAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(7000)
myColorAnimation.AutoReverse = true
myColorAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite
myBrush.ColorAnimations.Add(myColorAnimation)
anotherButton.Background = myBrush
```

SizeKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a new SizeKeyFrameCollection
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.

Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SizeKeyFrameCollection	
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.

IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

SizeModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this SizeModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified

	Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a SizeModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
SizeModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

SizeTimedM difier Class

Method	Descripti n
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this SizeModifier Inherited from SizeModifier.
Copy	Creates a copy of this SizeTimedModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from SizeModifier.
GetValueImpl	Inherited from SizeModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified

	object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a SizeModifier Inherited from SizeModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
SizeTimedModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.

End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is great than the simple duration of the animation, it will repeat itself until the time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

StringAnimationCollecti n Class
Definition: Represents a collection of StringModifier animations.

Method	Descripti n

Add	The Add(StringModifier) and Add(Object) methods add animations to the collection; the Add(String,StringAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified StringModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this StringAnimationCollection.
CopyTo	Copies the entire StringAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.
IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable,

	its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
StringAnimationCollection	Creates an empty StringAnimationCollection with a default capacity for a single animation.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty StringAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.

IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as StringAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of an StringAnimationCollection, the property calls the StringAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
      <LengthAnimation To="50" Duration="5" RepeatCount="500"
      AutoReverse="True"/>
    </LengthAnimationCollection>
   </Button.Width>

   A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
       <Button Canvas.Top="70" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Background>
    <SolidColorBrush Color="Blue">
     <SolidColorBrush.ColorAnimations>
      <ColorAnimation From="Red" To="Blue" Duration="7"
       RepeatCount="500" AutoReverse="True"/>
     </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

StringModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this StringModifier
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from

	Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a StringModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
StringModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Descripti n

AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

StringTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this StringTimedModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this StringModifier Inherited from StringModifier.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.

EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from StringModifier.
GetValueImpl	Inherited from StringModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a StringModifier Inherited from StringModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
StringTimedModifier	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	**Description**

Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.
CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current

	animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

Timeline Class
Definition: Maintains run-time timing state for timed objects.

Method	Description
BeginIn	Schedules a begin for some specified time in the future.
CloneCore	Clones this timeline.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline by disconnecting it from its current parent timeline.
DisableChildren	Disables the children of this timeline.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline by connecting it to its specified parent timeline. Auto-parented timelines can't be enabled with this method.
EndIn	Schedules an end for some specified time in the future.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	Returns an enumerator for the children of this container.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.

MakeUnchangeableCore	Called by the Changeable base class to ask the Timeline to make itself unchangeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
Pause	Pauses the timeline for this timeline and its children.
PropagateEventHandler	Called by the Changeable base class whenever the Changed event is hooked or unhooked.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Allows a timeline's timeline to progress again after a call to Pause.
Seek	Seeks a timeline's timeline to a new position.
Timeline	Creates a timeline with all timing attributes set to their default values.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the timeline plays in reverse after it completes its forward iteration.
Begin	Specifies the begin time for this timeline, in milliseconds.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	The current repeat period. The first period has a value of one.
CurrentTime	Gets the current position of the timeline, relative to the start time.
Deceleration	Specifies the percentage of a simple duration that is spent in the time deceleration phase.
Duration	Gets or sets the length of time the timeline takes to complete a

	single forward iteration, also known as the simple duration of an timeline.
End	Specifies the maximum end time for this timeline, in milliseconds.
EndSync	Defines how this container calculates its implicit duration.
Fill	The fill attribute for this timeline.
FillDefault	The default for the fill attribute for this timeline and its children.
HasChanged	Returns true if the timeline has changed since the last Tick.
IsAlive	True if the timeline can be started, false otherwise.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	True if the timeline is currently active, false otherwise.
IsEnabled	True if the timeline participates in a timing tree, false otherwise.
IsForwardProgressing	True if real time flows forward in this timeline.
IsOverridingBaseValue	True if the timeline is active or if it is past an active period but the fill attribute specifies that the state of the timeline should be preserved.
IsPaused	True if the timeline is currently paused, false otherwise.
IsReversed	True if this timeline is currently reversed, false otherwise.
ParentTimeline	Returns the intended parent timeline for the Timeline.
Progress	The current progress of time for this timeline.
RepeatCount	Gets or sets the number of times a timeline should repeat.
RepeatDuration	Gets or sets the total length of time a timeline should play. If this value is great than the simple duration of the timeline, it repeats itself until the time specified by this property.
Restart	Specifies the behavior of the timeline when a begin time is encountered after the first one (in any given activation period for the timeline's parent container).
RestartDefault	The default value for the Restart attribute for this timeline and its children.
Root	Returns the root timeline for the current UIContext.
Speed	Specifies the speed at which time runs in this timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

A timeline is used by animations and media to This class and its derived classes provide constructors that can be used to create timing objects. However, a timing object cannot be active until it is associated with a time manager. There are two ways to do this: by adding a timeline as a child of a time manager, or by adding a timeline to a container that is associated with a time manager.

TimelineBuilder Class
Definition: An object that can be used to create Timeline objects.

Meth d	Descripti n

Clear	Clears all properties previously set on this builder, so that the builder is reset to the same state it is at when it's first created.
CreateInstance	Creates a Timeline object that has the attributes currently set on this builder.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TimelineBuilder	Creates a new TimelineBuilder object.
ToString	Returns a String that represents the current Object. Inherited from Object.
ToTimeline	Creates a Timeline object that has the attributes currently set on this builder.

Property	Description
Acceleration	Accesses the Acceleration SMIL attribute.
AutoReverse	Accesses the AutoReverse SMIL attribute.
Begin	Accesses the Begin SMIL attribute.
Deceleration	Accesses the Deceleration SMIL attribute.
Duration	Accesses the Duration SMIL attribute.
End	Accesses the End SMIL attribute.
EndSync	Accesses the EndSync SMIL attribute.
Fill	Accesses the Fill SMIL attribute.
FillDefault	Accesses the FillDefault SMIL attribute.
ParentTimeline	
RepeatCount	Accesses the RepeatCount SMIL attribute.
RepeatDuration	Accesses the RepeatDuration SMIL attribute.
Restart	Accesses the Restart SMIL attribute.
RestartDefault	Accesses the RestartDefault SMIL attribute.
Speed	Accesses the Speed SMIL attribute.

TimelineBuilder objects are not thread-safe. The caller is responsible for ensuring that simultaneous access to a single TimelineBuilder object from two different threads does not occur.

TimeManager Class
Definition: The object that controls an entire timing tree.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing

	algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Restart	Resets the time manager and sets a new start time.
SetContext	Associates this UIContextObject with a UIContext. Inherited from UIContextObject.
Start	Starts the time manager at the current time.
Stop	Stops the timeline.
Tick	Moves the timeline forward to the current time and updates the state of all timing objects based on the time change.
TimeManager	Creates a time manager object in the stopped state.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
Clock	Accesses the reference clock used by this time manager to obtain real-world clock values.
Context	Returns the UIContext that this UIContextObject is associated with. Inherited from UIContextObject.
CurrentTime	The current position of the timeline, relative to the starting time. Setting this property to a new value has the effect of seeking the timing tree to a new point in time. Both forward and backward seeks are allowed. Setting this property has no effect if the timeline is stopped. However, seeking while the timeline is paused works as expected.
IsDirty	True if the structure of the timing tree has changed since the last tick.
RootTimeline	Returns the time container that is the root of the timing tree managed by this time manager.

A time manager controls the flow of time in a timing tree. The timeline is updated periodically by the rendering system, at which time the progress value of all active timelines is updated according to the elapsed time. This elapsed time can be controlled by the application by specifying a custom reference clock in the constructor.

TimeSyncValueTypeConverter Class
Definition: An object that performs type conversions involving TimeSyncValue values.

Method	Description
CanConvertFrom	Inherited from TypeConverter.
CanConvertFrom	Determines whether it is possible to convert an object of a given type into a TimeSyncValue value.
CanConvertTo	Inherited from TypeConverter.
CanConvertTo	Determines whether it is possible to convert a TimeSyncValue value into an object of a given type.
ConvertFrom	Converts an object into a TimeSyncValue value.
ConvertFrom	Inherited from TypeConverter.
ConvertFromInvariantString	Inherited from TypeConverter.
ConvertFromString	Inherited from TypeConverter.
ConvertTo	Inherited from TypeConverter.
ConvertTo	Converts a TimeSyncValue value into an object of a different type.

ConvertToInvariantString	Inherited from TypeConverter.
ConvertToString	Inherited from TypeConverter.
CreateInstance	Inherited from TypeConverter.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetConvertFromException	Inherited from TypeConverter.
GetConvertToException	Inherited from TypeConverter.
GetCreateInstanceSupported	Inherited from TypeConverter.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetProperties	Inherited from TypeConverter.
GetPropertiesSupported	Inherited from TypeConverter.
GetStandardValues	Inherited from TypeConverter.
GetStandardValuesExclusive	Inherited from TypeConverter.
GetStandardValuesSupported	Inherited from TypeConverter.
GetType	Gets the Type of the current instance. Inherited from Object.
IsValid	Inherited from TypeConverter.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SortProperties	Inherited from TypeConverter.
TimeSyncValueTypeConverter	
ToString	Returns a String that represents the current Object. Inherited from Object.

TimeTypeConverter Class
Definition: An object that performs type conversions involving Time values.

Method	Description
CanConvertFrom	Inherited from TypeConverter.
CanConvertFrom	Determines whether it is possible to convert an object of a given type into a Time value.
CanConvertTo	Inherited from TypeConverter.
CanConvertTo	Determines whether it is possible to convert a Time value into an object of a given type.
ConvertFrom	Converts an object into a Time value.
ConvertFrom	Inherited from TypeConverter.
ConvertFromInvariantString	Inherited from TypeConverter.
ConvertFromString	Inherited from TypeConverter.
ConvertTo	Inherited from TypeConverter.
ConvertTo	Converts a Time value into an object of a different type.
ConvertToInvariantString	Inherited from TypeConverter.

ConvertToString	Inherited from TypeConverter.
CreateInstance	Inherited from TypeConverter.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetConvertFromException	Inherited from TypeConverter.
GetConvertToException	Inherited from TypeConverter.
GetCreateInstanceSupported	Inherited from TypeConverter.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetProperties	Inherited from TypeConverter.
GetPropertiesSupported	Inherited from TypeConverter.
GetStandardValues	Inherited from TypeConverter.
GetStandardValuesExclusive	Inherited from TypeConverter.
GetStandardValuesSupported	Inherited from TypeConverter.
GetType	Gets the Type of the current instance. Inherited from Object.
IsValid	Inherited from TypeConverter.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SortProperties	Inherited from TypeConverter.
TimeTypeConverter	
ToString	Returns a String that represents the current Object. Inherited from Object.

VectorAnimation Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time. Inherited from VectorTimedModifier.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this VectorTimedModifier Inherited from VectorTimedModifier.
Copy	Creates a copy of this VectorModifier Inherited from VectorModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable. Inherited from VectorTimedModifier.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be

	retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null. Inherited from VectorTimedModifier.
EndIn	Schedules an interactive end time. Inherited from VectorTimedModifier.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	Inherited from VectorTimedModifier.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Calculates the value of the animation at the current time.
GetValueImpl	Inherited from VectorModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a VectorModifier Inherited from VectorModifier.
Pause	Pauses this timeline. Inherited from VectorTimedModifier.
PropagateEventHandler	PropagateEventHandler
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline. Inherited from VectorTimedModifier.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time. Inherited from VectorTimedModifier.
SetDefaultParentTimeline	
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an

	invalid state, this method throws an exception. Inherited from Changeable.
VectorAnimation	Initializes a new instance of the VectorAnimation class.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase. Inherited from VectorTimedModifier.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration. Inherited from VectorTimedModifier.
Begin	Gets or sets an offset to the start time of the animation. Inherited from VectorTimedModifier.
By	Gets or sets the total amount by which the animation changes its starting value.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation. Inherited from VectorTimedModifier.
CurrentTime	Gets the current time value of the animation. Inherited from VectorTimedModifier.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase. Inherited from VectorTimedModifier.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation. Inherited from VectorTimedModifier.
End	Gets or sets the maximum end time of the animation. Inherited from VectorTimedModifier.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set. Inherited from VectorTimedModifier.
Fill	Gets or sets a value that specifies the state of an object when its animation ends. Inherited from VectorTimedModifier.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines. Inherited from VectorTimedModifier.
From	Gets or sets the starting value of an animation.
InterpolationMethod	InterpolationMethod
IsAdditive	IsAdditive
IsChangeable	Gets a Boolean that indicates whether the object is currently

	modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from VectorTimedModifier.
IsCumulative	IsCumulative
IsEnabled	Inherited from VectorTimedModifier.
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future. Inherited from VectorTimedModifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from VectorTimedModifier.
IsPaused	Gets a value that indicates whether the animation is active and paused. Inherited from VectorTimedModifier.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline. Inherited from VectorTimedModifier.
KeyFrames	KeyValues
ParentTimeline	Gets or sets the default parent timeline of the animation. Inherited from VectorTimedModifier.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed. Inherited from VectorTimedModifier.
RepeatCount	Gets or sets the number of times an animation should repeat. Inherited from VectorTimedModifier.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property. Inherited from VectorTimedModifier.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached. Inherited from VectorTimedModifier.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines. Inherited from VectorTimedModifier.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline. Inherited from VectorTimedModifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline Inherited from VectorTimedModifier.
To	Gets or sets the ending value of the animation.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">

  <Button.Width>
    <LengthAnimationCollection>
     <LengthAnimation To="50" Duration="5" RepeatCount="500"
       AutoReverse="True"/>
    </LengthAnimationCollection>
  </Button.Width>

  A Button
 </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
  Height="30" Width="200">

  <Button.Background>
    <SolidColorBrush Color="Blue">
     <SolidColorBrush.ColorAnimations>
       <ColorAnimation From="Red" To="Blue" Duration="7"
         RepeatCount="500" AutoReverse="True"/>
     </SolidColorBrush.ColorAnimations>
    </SolidColorBrush>
  </Button.Background>

  Another Button
 </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.

This example demonstrates how to create an animation that repeats indefinitely. To make an animation repeat indefinitely in "XAML", set the animation's RepeatDuration property to Indefinite. In code, set the animation's RepeatDuration property to Time.Indefinite or set its RepeatCount property to double.PositiveInfinity.

In the following examples, a LengthAnimation is set to repeat indefinitely. Although this example uses a LengthAnimation, the procedure is the same for all the animation classes.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Line X1="10" Y1="20" X2="50" Y2="20"
  StrokeThickness="10" Stroke="Black">
  <Line.X2>
   <LengthAnimationCollection>
    <LengthAnimation From="30" To="300" Duration="10"
     RepeatDuration="Indefinite" />
   </LengthAnimationCollection>
  </Line.X2>
 </Line>

</Canvas>
```

```
        // C#
Line myLine = new Line();

LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.From = new Length(30);
myLengthAnimation.To = new Length(300);
myLengthAnimation.Duration = new Time(10000);
myLengthAnimation.RepeatDuration = Time.Indefinite;

LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

myLine.SetAnimations(Line.X2Property, collection);
```

```
        ' VB .NET
Dim myLine As new MSAvalon.Windows.Shapes.Line

Dim myLengthAnimation As new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.From = new MSAvalon.Windows.Length(30)
myLengthAnimation.To = new MSAvalon.Windows.Length(300)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(10000)
myLengthAnimation.RepeatDuration = _
 MSAvalon.Windows.Media.Animation.Time.Indefinite

Dim collection As new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

myLine.SetAnimations(Line.X2Property, collection)
```

Vect rAnimati nC llection Class
Definition: Represents a collection of VectorModifier animations.

Meth d	Description
Add	The Add(VectorModifier) and Add(Object) methods add animations to the collection; the Add(Vector,VectorAnimationCollection) method calculates the current value of the specified collection based on the specified base value.
AddChild	Implementation of AddChild. Adds a Modifier to this AnimationCollection from Markup.
AddText	Implementation of AddText. This is not implemented on this class.
Apply	Implementation of Apply. Applies an animation collection in markup to an element.
Clear	Clears the collection by setting the collection's Count to 0.
CloneCore	CloneCore
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Contains	Returns a Boolean that indicates whether the collection contains the specified VectorModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this VectorAnimationCollection.
CopyTo	Copies the entire VectorAnimationCollection to the specified one-dimensional array, starting at the specified index of the target array.
Disable	Inherited from AnimationCollection.
DisableImpl	
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Inherited from AnimationCollection.
EnableImpl	
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetEnumerator	GetEnumerator Inherited from AnimationCollection.
GetEnumeratorImpl	Returns an object that can be used to enumerate items in the list.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Returns the current value of the animation.
GetValueImpl	Provides a Modifier at a given index.

IndexOf	
Insert	
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Implementation of MakeUnchangeableCore.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	OnChanged
op_Addition	
op_Implicit	
PropagateEventHandler	Implementation of PropagateEventHandler.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Remove	
RemoveAt	
SetDefaultParentTimeline	Inherited from AnimationCollection.
SetValueImpl	Sets a Modifier at a given index.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
VectorAnimationCollection	Creates an empty VectorAnimationCollection with a default capacity for a single animation.
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AnimationType	AnimationType
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Inherited from AnimationCollection.
CountImpl	
Empty	An unchangeable empty VectorAnimationCollection.
IsChangeable	Gets a Boolean that indicates whether the object is currently

	modifiable. Inherited from Changeable.
IsChanging	Returns true if at least one of the animations in the animation list is currently active. Inherited from AnimationCollection.
IsFixedSize	
IsOverridingBaseValue	Returns true if at least one of the animations in the animation list is currently on. Inherited from AnimationCollection.
IsReadOnly	
IsSynchronized	
IsUsingBaseValue	Inherited from AnimationCollection.
IsUsingBaseValueImpl	
Item	this - typed version of indexer
Item	Use this to get or set a Modifier at a given index. Inherited from AnimationCollection.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
SyncRoot	
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

The AnimationCollection classes, such as VectorAnimationCollection, provide you with more control over how you animate a property than a single animation could. Each of the AnimationCollection classes effectively functions as a single composite animation, using all the animations in the collection to animate a base value. When a property requests the current value of a VectorAnimationCollection, the property calls the VectorAnimationCollection's GetValue method and passes it the property's base value. The first animation in the collection processes this base value and produces a result, which is then passed to the next animation in the collection, and so on, until the value has been processed by all the animations in the collection.

This example demonstrates how to animate a property using "Longhorn" markup language (code-named "XAML"). To animate a property, you associate the proper animation collection and animations with the property either directly or using the property's corresponding animation property. There are a variety of animation classes, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the RectangleWidth must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. Although in this example there is only one animation, you could associate multiple animations with a single property by placing the animations within the collection. Because there is no WidthAnimations property, the LengthAnimationCollection is associated directly with the button's Width property.

```
        <Canvas ID="root"
 xmlns="http://schemas.microsoft.com/2003/xaml">

 <Button Canvas.Top="20" Canvas.Left="20"
   Height="30" Width="200">
   <Button.Width>
     <LengthAnimationCollection>
       <LengthAnimation To="50" Duration="5" RepeatCount="500"
         AutoReverse="True"/>
     </LengthAnimationCollection>
   </Button.Width>

   A Button
  </Button>
```

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimation is used to animate the property. The SolidColorBrush.Color property has a corresponding ColorAnimations property, so the ColorAnimation is nested within the ColorAnimations property in order to animate the color of the brush.

```
        <Button Canvas.Top="70" Canvas.Left="20"
   Height="30" Width="200">

   <Button.Background>
     <SolidColorBrush Color="Blue">
       <SolidColorBrush.ColorAnimations>
         <ColorAnimation From="Red" To="Blue" Duration="7"
           RepeatCount="500" AutoReverse="True"/>
       </SolidColorBrush.ColorAnimations>
     </SolidColorBrush>
   </Button.Background>

   Another Button
  </Button>

</Canvas>
```

In the previous example, the ColorAnimationCollection tag, <ColorAnimationCollection>, is omitted when animating the brush's color. When animating a designated animation property—properties of the form PropertyNameAnimations, such as the ColorAnimations property in the previous example—you may omit the animation collection tag. However, when animating a property of a UI element—those classes that derive from UIElement—you must nest the animations within an animation collection tag.
For more information about animating properties, see Animation in "Avalon".

This example demonstrates how to animate a property in code. To animate a property, you associate the proper animation collection and animations with the property, either directly or using the property's corresponding animation property. There are a variety of animation classes in the MSAvalon.Windows.Media.Animation namespace, each of which animates a different kind of value.

Before some properties can be animated, they must be given a base value. For example, before animating the RectangleWidth of a Rectangle, the Rectangle must be set to a non-animated value (in this case, a Length object).

In the following example, the Width of a Button is animated. Because the Width property takes a Length, a LengthAnimation is needed. A LengthAnimationCollection is used to contain the LengthAnimation. After the animation is created, it is applied to the button's Width property using the SetAnimations method. When using the SetAnimations property, you pass it the static property identifier field (in this example

Button.WidthProperty) for the class and the AnimationCollection containing the animations. The code used to set the button's size and position has been omitted.

```
        // C#
Button aButton = new Button();

// Animate the Button's Width.
LengthAnimation myLengthAnimation = new LengthAnimation();
myLengthAnimation.To = new Length(50);
myLengthAnimation.Duration = new Time(5000);
myLengthAnimation.AutoReverse = true;
myLengthAnimation.RepeatDuration = Time.Indefinite;
LengthAnimationCollection collection = new LengthAnimationCollection();
collection.Add(myLengthAnimation);

// Set the animation.
aButton.SetAnimations(Button.WidthProperty, collection);
```

```
        ' VB .NET
Dim aButton As MSAvalon.Windows.Controls.Button
aButton = new MSAvalon.Windows.Controls.Button
MSAvalon.Windows.Controls.Canvas.SetLeft(aButton, new MSAvalon.Windows.Length(20))
MSAvalon.Windows.Controls.Canvas.SetTop(aButton, new MSAvalon.Windows.Length(20))
aButton.Width = new MSAvalon.Windows.Length(200)
aButton.Height = new MSAvalon.Windows.Length(30)
aButton.Content = "A Button"

' Animate the Button's Width.
Dim myLengthAnimation As MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation = new MSAvalon.Windows.Media.Animation.LengthAnimation
myLengthAnimation.To = new MSAvalon.Windows.Length(50)
myLengthAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(5000)
myLengthAnimation.AutoReverse = true
myLengthAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite
Dim collection As MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection = new MSAvalon.Windows.Media.Animation.LengthAnimationCollection
collection.Add(myLengthAnimation)

aButton.SetAnimations(MSAvalon.Windows.Controls.Button.WidthProperty, _
  collection)
```

In the previous example, the starting value of the animation isn't specified, so the animation uses the value of the button's Width as its starting value.

In the next example, the Background color of a second button is animated. The Background property takes a Brush. In this example, a SolidColorBrush is used to fill the button's Background, although a gradient, image, or pattern could have been used. To animate the button's background color, the Color of the SolidColorBrush must be animated. Because the SolidColorBrush.Color property accepts a Color, a ColorAnimationCollection and a ColorAnimation are used to animate the property. The Color property has a corresponding ColorAnimations property, so the ColorAnimation is added to the ColorAnimations property's ColorAnimationCollection using the Add method.

```
        // C#
//Create and set the second Button.
Button anotherButton = new Button();
```

```
// Create and animate a Brush to set the Button's
// background.
SolidColorBrush myBrush = new SolidColorBrush();
myBrush.Color = Colors.Blue;

ColorAnimation myColorAnimation = new ColorAnimation();
myColorAnimation.From = Colors.Blue;
myColorAnimation.To = Colors.Red;
myColorAnimation.Duration = new Time(7000);
myColorAnimation.AutoReverse = true;
myColorAnimation.RepeatDuration = Time.Indefinite;

myBrush.ColorAnimations.Add(myColorAnimation);
anotherButton.Background = myBrush;
```

```
        ' VB .NET
' Create and set the second Button.
Dim anotherButton As MSAvalon.Windows.Controls.Button
anotherButton = new MSAvalon.Windows.Controls.Button
Canvas.SetLeft(anotherButton, new MSAvalon.Windows.Length(20))
Canvas.SetTop(anotherButton, new MSAvalon.Windows.Length(70))
anotherButton.Width = new MSAvalon.Windows.Length(200)
anotherButton.Height = new MSAvalon.Windows.Length(30)
anotherButton.Content = "Another Button"

' Create and animate a Brush to set the Button's fill.
Dim myBrush As MSAvalon.Windows.Media.SolidColorBrush
myBrush = new MSAvalon.Windows.Media.SolidColorBrush
myBrush.Color = MSAvalon.Windows.Media.Colors.Blue
Dim myColorAnimation As MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation = new MSAvalon.Windows.Media.Animation.ColorAnimation
myColorAnimation.From = MSAvalon.Windows.Media.Colors.Blue
myColorAnimation.To = MSAvalon.Windows.Media.Colors.Red
myColorAnimation.Duration = new MSAvalon.Windows.Media.Animation.Time(7000)
myColorAnimation.AutoReverse = true
myColorAnimation.RepeatDuration = _
  MSAvalon.Windows.Media.Animation.Time.Indefinite
myBrush.ColorAnimations.Add(myColorAnimation)
anotherButton.Background = myBrush
```

VectorKeyFrameCollection Class

Method	Description
Add	Strongly typed implementation of Add.
CloneCore	Implementation of CloneCore.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a new VectorKeyFrameCollection

EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetCurrentSegmentValues	TODO
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	Makes a Changeable object immutable. Inherited from Changeable.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.
Validate	
ValidateObjectState	Implementation of ValidateObjectState.
VectorKeyFrameCollection	
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Pr perty	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help

	determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
Count	Implementation of Count.
Destination	The value specified in the last KeyFrame.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
Item	
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

VectorModifier Class

Method	Description
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this VectorModifier
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	
GetValueImpl	
IModifier.GetValue	Inherited from Modifier.

MakeUnchangeable	Makes an object immutable; after this method is called on a Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a VectorModifier
PropagateEventHandler	Shares a Changed event handler with the current object's data members or removes it. Inherited from Changeable.
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
VectorModifier	
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active. Inherited from Modifier.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period. Inherited from Modifier.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a

	complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.
UsesBaseValue	UsesBaseValue Inherited from Modifier.

VectorTimedModifier Class

Method	Description
BeginIn	Starts or restarts the animation at the specified offset from the current time.
CloneCore	Returns a modifiable shallow or deep clone of the current object. This abstract method must be implemented by classes that derive from Changeable. Inherited from Changeable.
CloneDownToUnchangeable	Returns an immutable copy of the specified object. Inherited from Changeable.
Copy	Creates a copy of this VectorModifier Inherited from VectorModifier.
Copy	Returns a modifiable copy of the current object. The copy's IsChangeable property is true and its StatusOfNextUse is Unchangeable. Inherited from Changeable.
Copy	Creates a copy of this VectorTimedModifier
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
EmbeddedChangeableReader	Accesses the specified Changeable data member, processes it, and returns a reference to the member. This reference should then be reassigned to the original member variable. Classes that derive from Changeable call this method on data members before they can be retrieved through property calls. Inherited from Changeable.
EmbeddedChangeableWriter	Processes a modified Changeable data member and returns a reference to the processed object. Inherited from Changeable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Schedules an interactive end time.
Equals	Determines whether two Object instances are equal. Inherited from Object.
FillInClone	
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
GetValue	Inherited from VectorModifier.
GetValueImpl	Inherited from VectorModifier.
IModifier.GetValue	Inherited from Modifier.
MakeUnchangeable	Makes an object immutable; after this method is called on a

	Changeable, its IsChangeable property is false. Inherited from Changeable.
MakeUnchangeableCore	MakeUnchangeableCore Inherited from Modifier.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ModifyHandlerIfChangeable	Adds or removes a Changed event handler to or from the specified Changeable object, if the object is currently modifiable. If the specified object is not modifiable—if its IsChangeable property is false—this method has no effect. Inherited from Changeable.
OnChanged	Called when the current object is modified. Classes that derive from Changed should call this method after they have been modified. Inherited from Changeable.
op_Implicit	Implicitly creates an AnimationCollection from a VectorModifier Inherited from VectorModifier.
Pause	Pauses this timeline.
PropagateEventHandler	
ReadPreamble	Ensures that simple (non-Changeable) members are being accessed from a valid user interface (UI) context. This method should be called before any simple members are accessed. Inherited from Changeable.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
Resume	Resumes this timeline.
Seek	Moves the current position of the animation backwards or forwards from either the current time, the Begin time, or the End time.
SetDefaultParentTimeline	SetDefaultParentTimeline Inherited from Modifier.
ToString	Returns a String that represents the current Object. Inherited from Object.
ValidateObjectState	Verifies that the current object has a valid state. If the object is in an invalid state, this method throws an exception. Inherited from Changeable.
VectorTimedModifier	
WritePostscript	Causes the current object to validate itself and then invokes the OnChanged method. Inherited from Changeable.
WritePreamble	Ensures that simple (non-Changeable) members are being accessed from a valid UI context. This method should be called before any simple members are set. Inherited from Changeable.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AllowChangeableReferenceOverride	Used in conjunction with the ChangeableUsageOverride type sent in as a parameter to ChangeableHelper.UseChangeable, to help determine when a Changeable being put into "use" should be promoted to "ChangeableReference". Inherited from Changeable.
AutoReverse	Gets or sets a value that indicates whether the animation plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the animation.
CanMakeUnchangeable	True if this Changeable can be made unchangeable. Inherited from Changeable.
CurrentRepeat	Gets the number of the current iteration of the animation.

CurrentTime	Gets the current time value of the animation.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time the animation takes to complete a single forward iteration, also known as the simple duration of an animation.
End	Gets or sets the maximum end time of the animation.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of an animation is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its animation ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current animation and its child timelines.
IsChangeable	Gets a Boolean that indicates whether the object is currently modifiable. Inherited from Changeable.
IsChanging	Gets a value that indicates whether the animation is active.
IsEnabled	
IsForwardProgressing	Gets a value that indicates whether the animation is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the animation is active or in a fill period.
IsPaused	Gets a value that indicates whether the animation is active and paused.
IsReversed	Gets a value that indicates whether the animation is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the animation.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times an animation should repeat.
RepeatDuration	Gets or sets the total length of time the animation should play. If this value is greater than the simple duration of the animation, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the animation's behavior when it is told to restart—that is, how the animation behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current animation and its child timelines.
Speed	Gets or sets the relative speed at which time should pass for the animation, compared to its parent timeline.
StatusOfNextUse	Gets or sets a UseStatus enumeration that specifies how the Changeable object behaves when it is "used." A Changeable object is considered used in the following situations: the object is set into a Property System property, the object is used as a sub-object in a complex Changeable object, or the object is used in a DrawingContext command. Inherited from Changeable.
Timeline	Timeline
UIContext	Gets the UIContext of the current object. The UIContext is used for maintaining thread safety. Inherited from Changeable.

UsesBaseValue	UsesBaseValue Inherited from Modifier.

INTERFACES

IClock Interface

Definition: Represents an object that can provide linear time values.

Property	Description
CurrentTime	Gets the current time in milliseconds.

The IClock interface must be implemented by any objects that wish to act as the clock driving the timing engine. This interface contains a single property that the time manager calls periodically to obtain the current time.

IModifier Interface

Definition: Defines the basic behavior of a modifier object. A modifier is an object that takes an object, called the base value, of a certain type and returns another object of the same type as its output.

Method	Description
GetValue	Calculates the modifier's current value from the passed base value and the internal state of the modifier. If the modifier's UsesBaseValue property is set to false, the modifier ignores the passed base value and calculates the current value from the internal state of the modifier.
SetDefaultParentTimeline	Sets the default parent Timeline of the modifier and its referenced timelines.

Property	Description
IsChanging	The method returns a Boolean indicating whether the modifier could be changing its value.
IsOverridingBaseValue	Gets a Boolean that indicates whether the modifier is overriding its base value.
UsesBaseValue	Returns a Boolean that indicates whether the modifier uses an external base value to calculate the result of the GetValue method.

ITimingControl Interface

Definition: Defines the behavior of timelines and timed objects.

Method	Description
BeginIn	Starts or restarts the timeline at the specified offset from the current time.
Disable	Disables this timeline, after which the timeline can no longer become active. The timeline can be re-enabled with a call to Enable.
Enable	Enables this timeline, parenting it to the timeline specified by the ParentTimeline property. This allows the timeline to become active. This method throws an exception if the ParentTimeline property is null.
EndIn	Ends the timeline at the specified offset from the current time.
Pause	Pauses the timeline and all its child timelines.
Resume	Unpauses the timeline and all its child timelines.

Seek	Moves the current position of a timeline backwards or forwards from either the current time, the Begin time, or the End time.

Property	Description
Acceleration	Gets or sets the fraction of the simple duration spent in the acceleration phase.
AutoReverse	Gets or sets a value that indicates whether the timeline plays in reverse after it completes its forward iteration.
Begin	Gets or sets an offset to the start time of the timeline.
CurrentRepeat	Gets the number of the current iteration of the timeline.
CurrentTime	Gets the current time value of the timeline.
Deceleration	Gets or sets a value that represents the fraction of the simple duration spent in the deceleration phase.
Duration	Gets or sets the length of time a timeline takes to complete a single forward iteration, also known as the simple duration of an timeline.
End	Gets or sets the maximum end time of the timeline.
EndSync	Gets or sets a TimeEndSync enumeration that specifies how the implicit duration of a timeline is determined. This property is only used if the Duration property is not explicitly set.
Fill	Gets or sets a value that specifies the state of an object when its timeline ends.
FillDefault	Gets or sets a value that indicates the default value of the Fill property of the current timeline and its child timelines.
IsChanging	Gets a value that indicates whether the timeline is active.
IsEnabled	Gets a value that indicates whether the timeline is part of a timing sub-tree.
IsForwardProgressing	Gets a value that indicates whether the timeline is progressing from past to future.
IsOverridingBaseValue	Gets a value that indicates whether the timeline is active or in a fill period.
IsPaused	Gets a value that indicates whether the timeline is active and paused.
IsReversed	Gets a value that indicates whether the timeline is currently moving in the opposite direction of its parent timeline.
ParentTimeline	Gets or sets the default parent timeline of the current timeline.
Progress	Gets a number from 0 to 1 that indicates the fraction of the simple duration that has elapsed.
RepeatCount	Gets or sets the number of times a timeline should repeat.
RepeatDuration	Gets or sets the total length of time a timeline should play. If this value is greater than the simple duration of the timeline, it will repeat itself for the length of time specified by this property.
Restart	Gets or sets the timeline's behavior when it is told to restart—that is, how the timeline behaves when a second begin time is reached.
RestartDefault	Gets or sets the default value of the Restart property of the current timeline and its children.
Speed	Gets or sets the relative speed at which time should pass for this timeline, compared to its parent timeline.

ITimingControlBuilder Interface
Definition: Represents an object that can build a timeline template.

Pr perty	Descripti n
Acceleration	Accesses the Acceleration SMIL attribute.

AutoReverse	Accesses the AutoReverse SMIL attribute.
Begin	Accesses the Begin SMIL attribute.
Deceleration	Accesses the Deceleration SMIL attribute.
Duration	Accesses the Duration SMIL attribute.
End	Accesses the End SMIL attribute.
EndSync	Accesses the EndSync SMIL attribute.
Fill	Accesses the Fill SMIL attribute.
FillDefault	Accesses the FillDefault SMIL attribute.
ParentTimeline	Accesses the ParentTimeline attribute.
RepeatCount	Accesses the RepeatCount SMIL attribute.
RepeatDuration	Accesses the RepeatDuration SMIL attribute.
Restart	Accesses the Restart SMIL attribute.
RestartDefault	Accesses the RestartDefault SMIL attribute.
Speed	Accesses the Speed SMIL attribute.

MSAvalon.Windows.Media.TextFormatting

The following tables list the members exposed by the MSAvalon.Windows.Media.TextFormatting namespace.

Classes

InlineObjectInfo	Provides measurement details for inline text objects. The formatting client passes this object as a parameter to the GetInlineObjectInfo method.
TextFormatter	TextFormatter is the "Avalon" text engine and provides services for formatting text and breaking text lines. TextFormatter can handle different text character formats and paragraph styles, and includes support for international text layout.
TextHighlightBounds	Bounds of text range
TextInfo	Represents information about a block of text in the client's text source character store.
TextLine	Provides services to a line of text. Inherit from this class to implement services that manipulate and format a line of text. This is an abstract class.
TextMarkerGeneratedContent	Generates line list marker output.
TextMarkerInfo	Defines the style and type of a paragraph's list marker. The formatting client uses this class as a parameter to provide marker details to the GetTextMarkerInfo method.
TextParagraphProperties	Represents properties that can change from one paragraph to the next, such as flow direction, alignment, or indentation.
TextRun	Defines a sequence of characters that share a single property set. The formatting client provides TextRun details into this class when the TextFormatter passes it as a parameter to the GetTextRun method.
TextRunBounds	Bounds of text run
TextRunCache	Provides caching services to the TextFormatter object in order to improve performance.
TextRunClientData	Represents client information data associated with a TextRun.
TextRunProperties	Provides properties that can change from one TextRun to another, such as typeface or foreground brush. This is an abstract class.
TextRunTypographyProperties	Provides typography properties for TextRun. This client set of properties generates a set of features that are processed by the OpenType layout engine.
TextSource	Provides character data and formatting properties to the TextFormatter. All access to the text in the TextSource is achieved through the GetTextRun method, which is designed to allow the client to virtualize text in any way it chooses. This is an abstract class.
TextTrimmingInfo	Provides description of text trimming characteristics. The formatting client fills trimming details into this class when the TextFormatter passes it as a parameter of the GetTextTrimmingInfo method.

Enumerations

TextParagraphFlags	Flags describing paragraph characteristics
TextRunCacheFlags	Kind of content in text run cache
TextRunType	Indicates the type of TextRun.

Structures

TextSourceCharacterIndex	TextSourceCharacterIndex represents a caret or character position in text.

MSAvalon.Windows.Media.TextFormatting

CLASSES

InlineObjectInf Class

Definition: Provides measurement details for inline text objects. The formatting client passes this object as a parameter to the GetInlineObjectInfo method.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.

TextFormatter Class

Definition: TextFormatter is the "Avalon" text engine and provides services for formatting text and breaking text lines. TextFormatter can handle different text character formats and paragraph styles, and includes support for international text layout.

Method	Description
Create	Gets the **TextFormatter** object of the current running thread.
Dispose	Disposes of obsolete **TextFormatter** internal resources.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
FormatLine	Creates a TextLine that is used to format and display a document.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextFormatter	Initializes a new instance of **TextFormatter**.
ToString	Returns a String that represents the current Object. Inherited from Object.

Unlike a traditional text application programming interface (API), the **TextFormatter** interacts with a text layout client through a set of callback methods. It requires the client to provide these methods in an implementation of the TextSource class.

TextHighlightB unds Class
Definition: Bounds of text range

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetTextRunBounds	Get array of rectangles each corresponding to rectangle bounds of each run
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
FlowDirection	Text flow direction inside the boundary rectangle
Rectangle	Bounds rectangle

TextInfo Class
Definition: Represents information about a block of text in the client's text source character store.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetCharacterBuffer	Sets the **TextInfo** character buffer.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
CharacterOffset	Sets the character offset relative to the beginning of the characterBuffer array to the first character of the TextRun.
Length	Gets or sets a value that represents the number of characters in the TextSource character store.

TextLine Class
Definition: Provides services to a line of text. Inherit from this class to implement services that manipulate and format a line of text. This is an abstract class.

Method	Descripti n
Dispose	Disposes of an obsolete instance of TextLine.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetDistanceFromTextSourceCharacterIndex	Gets the distance from the beginning of a line of text from the specified textSourceCharacterIndex.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetSubLineCollection	Gets a collection of glyph runs that constitute the current line.
GetTextHighlightBounds	Client to get an array of bounding rectangles of a range of characters within a text line.
GetTextRunClientDataCollection	Gets a collection of client-specific objects associated with each TextRun in the line.
GetTextSourceCharacterIndexFromDistance	Gets the TextSource character index that corresponds to the specified distance from the beginning of the line.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextLine	Initializes a new instance of TextLine.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
Baseline	Gets the distance from the top to the baseline of the current TextLine.
DependantCharacterLength	Gets the number of characters that follow the last character of the line. This information may cause the current line to be reformatted.
EmergencyWrapped	Client to get a boolean value indicates whether the text line is emergency wrapped at the location nearest to the margin due to lack of valid linebreak opportunity in the line.
Extent	Gets the distance from the topmost to bottommost black pixel in a line.
Height	Gets the height of a line of text.
Length	Gets the total number of TextSource positions of the current line.
MarkerBaseline	Gets the distance from the edge of the line's highest point to the baseline marker of the line.
MarkerHeight	Gets the height of a list item's marker.
MinWidth	Gets the minimum width possible for a given line.
OverhangAfter	Gets the distance that black pixels extend beyond the bottom alignment edge of a line.

OverhangLeading	Gets the distance that black pixels extend prior to the left leading alignment edge of the line.
OverhangTrailing	Gets the distance that black pixels extend following the right trailing alignment edge of the line.
Start	Gets the distance from the start of a paragraph to the starting point of a line.
TrailingWhitespaceLength	Gets the number of whitespace code points beyond the last non-blank character in a line.
Width	Gets the width of a line of text without trailing whitespaces.
WidthIncludingTrailingWhitespace	Gets the width of a line of text, including trailing whitespaces.

TextMarkerGeneratedContent Class
Definition: Generates line list marker output.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextMarkerGeneratedContent	Initializes a new instance of TextMarkerGeneratedContent.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
Properties	Gets or sets run properties of the TextMarkerGeneratedContent.
UnicodeString	Gets or sets a unicode string used for accessibility.

TextMarkerInfo Class
Definition: Defines the style and type of a paragraph's list marker. The formatting client uses this class as a parameter to provide marker details to the GetTextMarkerInfo method.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.

MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
AutoNumberingIndex	Gets or sets an index number for a counter-style list marker.
Offset	Gets or sets the distance from the text to the end of a list marker.
RunProperties	Gets or sets text properties used to render the list marker.
Type	Gets or sets the style of a text paragraph's list marker.

TextParagraphProperties Class
Definition: Represents properties that can change from one paragraph to the next, such as flow direction, alignment, or indentation.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextParagraphProperties	Initializes a new instance of TextParagraphProperties.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
DefaultIncrementalTab	Gets or sets the default incremental tab distance.
DefaultTextRunProperties	Gets or sets the default TextRunProperties.
Flags	Gets or sets the Flags that describe characteristics of the paragraph of text.
FlowDirection	This property specifies whether the primary text advance direction shall be left-to-right, right-to-left, or top-to-bottom.
HorizontalAlignment	This property describes how inline content of a block is aligned.
Hyphenator	Gets or sets access to the Hyphenation service provider interface.
Indent	Gets or sets the amount of line indentation.
LineHeight	Gets or sets the height of a line of text.
TextWrap	Gets or sets the means by which text wraps when it reaches the edge of its containing box.

TextRun Class
Definition: Defines a sequence of characters that share a single property set. The formatting client provides TextRun details into this class when the TextFormatter passes it as a parameter to the GetTextRun method.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
SetCharacterBuffer	Sets the buffer type of the TextRun.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
CharacterOffset	Gets or sets the character offset relative to the beginning of the TextRun array to the first character of the run.
ClientData	Gets or sets the TextFormatter object that the client then assigns to the TextRun.
Length	Gets or sets the number of TextSource character indices that the TextRun occupies.
Properties	Gets or sets text properties that can change from one TextRun to the next, such as typeface or foreground brush.
Type	Gets or sets the type of TextRun.

TextRunBounds Class
Definition: Bounds of text run

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
CharacterLength	character length of bounded text run
Rectangle	Text run bounding rectangle

TextSourceCharacterIndex	First text source character index of text run
Type	type of the run

TextRunCache Class
Definition: Provides caching services to the TextFormatter object in order to improve performance.

Method	Description
Change	Notifies the client of a change in the cache when text or properties of the TextRun are added, removed or replaced.
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
InvalidateCharacterBufferReferences	Notifies the client of possible changes in the character buffer or character offset of the TextRun due to changes occuring in other parts of the text store.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextRunCache	Initializes a new instance of TextRunCache.
ToString	Returns a String that represents the current Object. Inherited from Object.

Applications that use this caching mechanism are responsible for invalidating the content in the cache when it has changed.

TextRunClientData Class
Definition: Represents client information data associated with a TextRun.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
ToString	Returns a String that represents the current Object. Inherited from Object.

Pr perty	Descripti n
ClientData	Gets or sets client-specific data associated with a TextRun.
Length	Gets or sets the character length of a TextRun.
Type	Gets or sets the type of the TextRun.

TextRunProperties Class
Definition: Provides properties that can change from one TextRun to another, such as typeface or foreground brush. This is an abstract class.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextRunProperties	Initializes a new instance of TextRunProperties.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
FontHintingEmSize	Gets or sets the text size in points, which is then used for purposes of font hinting.
FontRenderingEmSize	Gets or sets the text size in points.
ForegroundBrush	Gets or sets the foreground brush of the TextRun.
HighlightBackgroundBrush	Gets or sets the brush for the highlight background of the TextRun.
LocaleID	Gets the locale ID of the TextRun.
TextDecorationBrush	Gets or sets the Brush used for text decoration.
TextDecorationPen	Gets or sets the Pen used for text decoration.
TextDecorations	Gets or sets the TextDecorations used for the TextRun.
Typeface	Gets or sets the typeface of the TextRun.
TypographyProperties	Gets or sets the typography properties of the TextRun.
VerticalBoxAlignment	Gets or sets the vertical box alignment of the TextRun.

The client provides a concrete implementation of this abstract class. This enables the client to implement run properties in a way that fits with their formatting store.

TextRunTypographyProperties Class
Definition: Provides typography properties for TextRun. This client set of properties generates a set of features that are processed by the OpenType layout engine.

Meth d	Description
Equals	Determines whether two Object instances are equal. Inherited from

	Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
OnPropertiesChanged	Should be called every time any property changes it's value
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextRunTypographyProperties	Initializes a new instance of TextRunTypographyProperties.
ToString	Returns a String that represents the current Object. Inherited from Object.

Property	Description
AnnotationAlternates	Gets or sets a value that specifies whether alternate notation forms are enabled. These forms include glyphs placed in open or solid circles, squares, parentheses, diamonds or rounded boxes.
Capitals	Gets or sets a value that indicates the capital form of the selected font. Capital forms include Normal, SmallCaps, AllSmallCaps, PetiteCaps, AllPetiteCaps, Unicase and Titling.
CapitalSpacing	Gets or sets a value that indicates whether inter-glyph spacing for all-capital text is globally adjusted to improve readability.
ContextualAlternates	Gets or sets a value that indicates whether custom glyph forms can be used based upon the context of the text being rendered. This is valuable for languages that have letters with multiple medial forms that are used depending on the context of the surrounding letters.
ContextualLigatures	Gets or sets a value that indicates whether contextual ligatures are supported.
ContextualSwashes	Gets a value that indicates contextual swashes are enabled. Contextual swashes are typographic forms that are only applied based on the context of the text being rendered.
DiscretionaryLigatures	Gets or sets a value that indicates whether discretionary ligatures are enabled. Discretionary ligatures are typographic forms that font designers may add to their fonts to assist with readability.
EastAsianExpertForms	Gets or sets a value that indicates whether standard Japanese font forms have been replaced with the corresponding preferred typographic form.
EastAsianLanguage	Gets or sets a value that specifies the version of glyphs to be used for a specific writing system or language.
EastAsianWidths	Gets or sets a value that indicates the proportional width to be used for Latin characters in an East Asian font.
Fraction	Gets or sets a value that indicates whether special fractional glyph forms are available.
HistoricalForms	Gets or sets a value that indicates whether historical font forms are available. Historical forms are typographic conventions that were common in the past such as the long form of "s" or the Fraktur "k".
HistoricalLigatures	Gets or sets a value that indicates whether historical ligatures are enabled. Historical ligatures are typographic forms used in historical typography that font designers may add to their fonts.

JustificationAlternates	Gets or sets a value that indicates whether alternate glyphs of varying widths may be substituted to justify text.
Kerning	Gets or sets a value that indicates whether kerning is enabled. Kerning is a typographic function that adjusts the spacing between characters to enhance word shape.
MathematicalGreek	Gets or sets a value that indicates whether standard typographic font forms of Greek glyphs have been replaced with corresponding font forms commonly used in mathematical notation.
NumeralAlignment	Gets or sets the glyph set to be used for different numeric alignment scenarios.
NumeralStyle	Gets or sets a value that selects a set of glyphs that are used to render numeric alternate font forms that better match with their associated text.
SlashedZero	Gets or sets a value that indicates whether a nominal zero font form should be replaced with a slashed zero.
StandardLigatures	Gets or sets a value that indicates whether standard ligatures are enabled. Standard ligatures assist with readability.
StandardSwashes	Gets or sets a value that indicates if standard swashes are enabled. Standard swashes are typographic forms that font designers may add to their fonts that have common usage.
StylisticAlternates	Gets or sets a value that indicates whether stylistic alternates are enabled. Stylistic alternates are glyph designs for a purely aesthetic effect that don't fit into a category such as historical or swash.
StylisticSet1	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet10	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet11	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet12	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet13	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet14	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet15	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.
StylisticSet16	Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a

character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet17 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet18 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet19 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet2 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet20 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet3 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet4 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet5 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet6 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet7 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet8 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

StylisticSet9 Gets or sets a value that indicates whether a stylistic set of a font form is enabled. Some fonts contain stylistic variant glyphs that correspond to portions of a character set. Glyphs in stylistic sets may be designed to harmonize visually, interact in particular ways, or work together in other ways.

Variants Gets or sets a value that indicates a variation of the standard typographic form should be used. Variants are similar to superscript or subscript font forms. It is possible for a font form to have glyph sets that differ between superscript and ordinal forms, or between subscript and inferior forms.

TextSource Class
Definition: Provides character data and formatting properties to the TextFormatter. All access to the text in the TextSource is achieved through the GetTextRun method, which is designed to allow the client to virtualize text in any way it chooses. This is an abstract class.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetInlineObjectInfo	TextFormatter to get metrics of inline object started at specified text source character index
GetPrecedingText	Instructs the TextFormatter to get the text immediately before the specified TextSource position.
GetText	Instructs the TextFormatter to get the text at a specified TextSource position.
GetTextMarkerInfo	TextFormatter to get marker information of text paragraph
GetTextRun	Instructs TextFormatter to get a TextRun starting at a specified TextSource position.
GetTextTrimmingInfo	TextFormatter to get text trimming characteristics when text overflows the margin.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.
TextSource	Initializes a new instance of TextSource.
ToString	Returns a String that represents the current Object. Inherited from Object.

TextTrimmingInfo Class
Definition: Provides description of text trimming characteristics. The formatting client fills trimming details into this class when the TextFormatter passes it as a parameter of the GetTextTrimmingInfo method.

Method	Description
Equals	Determines whether two Object instances are equal. Inherited from Object.
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. Inherited from Object.
GetHashCode	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. Inherited from Object.
GetType	Gets the Type of the current instance. Inherited from Object.
MemberwiseClone	Creates a shallow copy of the current Object. Inherited from Object.
ReferenceEquals	Determines whether the specified Object instances are the same instance. Inherited from Object.

ToString	Returns a String that represents the current Object. Inherited from Object.

Pr perty	Description
EllipsisBackgroundBrush	Gets or sets the background brush of a trimming ellipsis symbol.
EllipsisForegroundBrush	Gets or sets the foreground brush of a trimming ellipsis symbol.
Type	Gets or sets the style of trimming.